



TUGAS AKHIR TF 145565

**RANCANG BANGUN SISTEM *MONITORING*
KONSENTRASI GAS KARBON MONOKSIDA (CO)
SEBAGAI EMISI GAS BUANG MENGGUNAKAN SENSOR
GAS MQ-7 BERBASIS MIKROKONTROLER STM32F4
*DISCOVERY***

**Lailatul Mufida
NRP. 2414 031 062**

**Dosen Pembimbing
Dr.Ir.Ali Musyafa', MSc
NIP. 19600901 198701 1 001**

**PROGRAM STUDI D3 TEKNIK INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



TUGAS AKHIR TF 145565

**RANCANG BANGUN SISTEM *MONITORING*
KONSENTRASI GAS KARBON MONOKSIDA (CO)
SEBAGAI EMISI GAS BUANG MENGGUNAKAN
SENSOR GAS MQ-7 BERBASIS MIKROKONTROLER
STM32F4 *DISCOVERY***

**Lailatul Mufida
NRP. 2414 031 062**

**Dosen Pembimbing
Dr.Ir.Ali Musyafa', MSc
NIP. 19600901 198701 1 001**

**PROGRAM STUDI D3 TEKNIK INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



FINAL PROJECT – TF 145565

***DESIGN CONSTRUCTION MONITORING CONCENTRATION
CARBON MONOXIDE (CO) GAS AS GAS EMISSION WASTE
USING SENSOR GAS MQ 7 BASED ON STM32F4
DISCOVERY***

**Lailatul Mufida
NRP. 2414 031 062**

Advisor Lecturer
Dr.Ir.Ali Musyafa', MSc
NIP. 19600901 198701 1 001

***STUDY PROGRAM OF D3 OF INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2016***

**RANCANG BANGUN SISTEM *MONITORING*
KONSENTRASI GAS KARBON MONOKSIDA (CO)
SEBAGAI EMISI GAS BUANG MENGGUNAKAN
SENSOR GAS MQ-7
BERBASIS MIKROKONTROLER STM32F4 *DISCOVERY***

TUGAS AKHIR

**Oleh :
Lailatul Mufida
NRP. 2414 031 062**

Surabaya, 25 Juli 2017

**Menyetujui,
Dosen Pembimbing**

**Dr. Ir. Ali Musyafa', M.Sc.
NIP. 19600901 198701 1 001**

**Mengetahui,
Ketua Departemen
Teknik Instrumentasi FV-ITS,**



**Dr. Ir. Purwadi Agus Darwito, M.Sc.
NIP. 19620822 198803 1 001**

**RANCANG BANGUN SISTEM *MONITORING*
KONSENTRASI GAS KARBON MONOKSIDA (CO)
SEBAGAI EMISI GAS BUANG MENGGUNAKAN
SENSOR GAS MQ-7
BERBASIS MIKROKONTROLER STM32F4 *DISCOVERY***

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada
Program Studi DIII Teknik Instrumentasi
Jurusan Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

Lailatul Mufida
NRP. 2414 031 062

Disetujui oleh Tim Penguji Tugas Akhir :

- | | | |
|----------------------------------------|-------|---------------------|
| 1. Dr. Ir. Ali Musyafa', M.Sc. | | (Dosen Pembimbing) |
| 2. Dr. Ir. Purwadi Agus Darwito, M.Sc. | | (Ketua Tim Penguji) |
| 3. Detak Yan Pratama S.T. M.Sc. | | (Dosen Penguji 1) |

SURABAYA
25 JULI 2017

**RANCANG BANGUN SISTEM *MONITORING* KONSENTRASI
GAS KARBON MONOKSIDA (CO) SEBAGAI EMISI GAS
BUANG MENGGUNAKAN SENSOR GAS MQ-7
BERBASIS MIKROKONTROLER STM32F4 *DISCOVERY***

Nama Mahasiswa : Lailatul Mufida
NRP : 2414 031 062
Program Studi : D III Teknik Instrumentasi
Departemen : Teknik Instrumentasi
Fakultas : Fakultas Vokasi
Dosen Pembimbing : Dr.Ir.Ali Musyafa', MSc

Abstrak

Gas karbon monoksida adalah gas yang dihasilkan oleh emisi gas buang kendaraan bermotor yang menggunakan bahan bakar bensin, dan pembakaran minyak dari batubara, industri dan sampah domestik. Nilai ambang batas karbon monoksida (CO) sebesar 25 ppm. Apabila manusia menghirup gas CO, maka hemoglobin dan CO akan berikatan dan dapat mengganggu kesehatan manusia. Sehingga dalam tugas akhir ini perlu adanya sebuah rancang bangun sistem monitoring gas karbonmonoksida (CO) berbasis STM32F4 *Discovery* untuk memonitoring konsentrasi gas karbonmonoksida (CO) khususnya di wilayah Surabaya. Sistem monitoring gas karbonmonoksida ini memiliki spesifikasi alat dengan *range* sebesar 6,93-32,23 ppm, *span* sebesar 25,3, resolusi sebesar 0,01, non-linieritas sebesar 2,96%, akurasi sebesar 84%, dan kesalahan (*error*) sebesar 16%. Nilai hasil pengukuran $x = 32,23 \pm 1,27$ (ppm), *confidence level* (CL) 95%, dan faktor cakupan (k) sebesar $\pm 2,171$. Sistem monitoring ini dilengkapi penyimpanan *data logger* menggunakan *SD Card* dan komunikasi antar *mobile* menggunakan *SMS Gateway*.

Kata Kunci: Gas Karbon monoksida (CO), Sensor Gas MQ-7, Spesifikasi Alat, Hasil Pengukuran, *Data Logger*, *SMS Gateway*

**DESIGN CONSTRUCTION MONITORING
CONCENTRATION CARBON MONOXIDE (CO) GAS AS
GAS EMISSION WASTE USING SENSOR GAS MQ 7
BASED ON STM32F4 DISCOVERY**

Name of Student : Lailatul Mufida
NRP : 2414 031 062
Program Study : D III Instrumentation Engineering
Department : Instrumentation Engineering
Faculty : Faculty of Vocation
Advisor Lecturer : Dr.Ir.Ali Musyafa', MSc

ABSTRACT

Carbon monoxide is a gas produced by motor vehicle exhaust emissions that use gasoline, and burning oil from coal, industrial and domestic waste. Carbon monoxide (CO) threshold value of 25 ppm. When humans inhale CO gas, then hemoglobin and CO will bind and can disrupt human health. So, in this final project, it is necessary to design a monitoring system of carbonmonoxide gas (CO) based on STM32F4 Discovery to monitor the concentration of carbon monoxide (CO) gas especially in Surabaya area. This carbon monoxide gas monitoring system has a device specification ranging from 6.93 to 32.23 ppm, span of 25.3, a resolution of 0.01, non-linearity of 2.96%, accuracy of 84%, and error of 16%. The measured values $x = 32.23 \pm 1.27$ (ppm), confidence level (CL) 95%, and coverage factor (k) of ± 2.171 . This monitoring system is equipped with data storage logger using SD Card and communication between mobile using SMS Gateway.

Keywords: Carbon monoxide gas (CO), Gas Sensor MQ-7, Specification, Measurement Results, Data Logger, SMS Gateway

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“RANCANG BANGUN SISTEM *MONITORING* KONSENTRASI GAS KARBON MONOKSIDA (CO) SEBAGAI EMISI GAS BUANG MENGGUNAKAN SENSOR GAS MQ-7 BERBASIS MIKROKONTROLER STM32F4 *DISCOVERY*”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan dan peran dari orangtua dan keluarga besar serta berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Ketua Program Studi D3 Metrologi dan Instrumentasi.
2. Bapak Dr.Ir.Ali Musyafa', MSc selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Bapak Fitri Adi Iskandariato, ST, MT selaku Dosen Wali penulis.
4. Bapak dan Ibu tercinta orang tua terbaik dan yang paling saya sayangi berkat doanya yang membuat saya bisa menyelesaikan tugas akhir ini.
5. Seluruh Asisten Laboratorium Pengukuran Fisis yang telah membantu dalam pengerjaan Tugas Akhir penulis.
6. Mas Ahmadi, Mas Bagus, dan Mas Okik selaku mentor dalam konsultasi penggunaan *STM32F4 Discovery*
7. *Monitoring Gas Team*, sahabat terbaik yang penulis cintai (Atik Sinawang W, Syahril Arisdianta, Haryo Arif) yang telah bersama-sama berjuang dalam pengerjaan Tugas Akhir ini hingga selesai.
8. Kakak tercinta, Mas Itak yang memberikan dukungan dan doa untuk menyelesaikan Tugas Akhir ini.
9. Mas Kimi yang memberikan semangat, doa dan kepercayaan untuk menyelesaikan Tugas Akhir ini.
10. Teman-teman D3 Metrologi & Instrumentasi dan S1 Teknik Fisika angkatan 2013 FTI-ITS.
11. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih kurang sempurna. Oleh karena itu penulis menerima segala masukan baik berupa saran, kritik, dan segala bentuk tegur sapa demi kesempurnaan laporan ini.

Demikian laporan Tugas Akhir ini penulis persembahkan dengan harapan dapat bermanfaat dalam akademik baik bagi penulis sendiri maupun bagi pembaca.

Surabaya, 25 Juli 2017

Penulis.

DAFTAR ISI

	Hal
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN I.....	iii
LEMBAR PENGESAHAN II.....	iv
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xv

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Manfaat.....	4
1.6 Sistematika Laporan	4

BAB II TINJAUAN PUSTAKA

2.1 Gas Karbon Monoksida (CO).....	5
2.2 Emisi Gas Buang Karbon Monoksida (CO)	6
2.3 Nilai Ambang Batas.....	6
2.4 Tanda dan Gejala Keracunan Karbon Monoksida (CO).....	7
2.5 Sensor Gas MQ-7	9
2.6 Mikrokontroler.....	11
2.7 Mikrokontroler STM32F4 <i>Discovery</i>	11
2.8 LCD 16X4	13
2.9 <i>RTC</i> DSS3231.....	15
2.10 <i>Micro SD Shield</i> Modul	16
2.11 <i>Data Logger</i>	16
2.12 <i>Modem GSM Wavcom M1306B Fasttrack</i>	17
2.13 RS232	19

2.14 <i>Fan DC</i>	21
2.15 <i>Power Supply 12VDC 2A</i>	22
2.16 <i>MCB (Miniature Circuit Breaker)</i>	22
2.17 <i>IC 7805</i>	23
2.18 <i>ChibiOs</i>	24
2.19 <i>Qt Creator</i>	25
2.20 <i>STM32 ST-LINK</i>	26
2.21 Analisa Sumber-Sumber Ketidakpastian.....	27

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 <i>Flowchart dan Diagram Blok Perancangan Alat</i>	33
3.2 <i>Desain Rancang Bangun Sistem Monitoring Gas CO</i>	35
3.3 <i>Perancangan Sensor Gas MQ-7</i>	36
3.4 <i>Perancangan Shield Board STM32F4 Discovery</i>	38
3.5 <i>Perancangan Liquid Crystal Display (LCD)</i>	41
3.6 <i>Perancangan Real Time Clock (RTC)</i>	42
3.7 <i>Perancangan Modul SD Card</i>	43
3.8 <i>Perancangan Modem Wavecom</i>	44
3.9 <i>Perancangan Power Supply</i>	46
3.10 <i>Perancangan dan Pembuatan Software</i>	47

BAB IV PENGUJIAN DAN ANALISIS DATA

4.1 <i>Rancang Bangun Alat</i>	55
4.2 <i>Pengujian Real Time Clock (RTC)</i>	56
4.3 <i>Pengujian Penyimpanan pada SD Card</i>	57
4.4 <i>Pengujian Modem Wavecom pada SMS Gateway</i>	58
4.5 <i>Pengujian Rangkaian Sensor MQ 7</i>	60
4.6 <i>Data Spesifikasi Alat</i>	66
4.7 <i>Data Hasil Kalibrasi</i>	68
4.8 <i>Pembahasan</i>	73

BAB V PENUTUP

5.1 <i>Kesimpulan</i>	75
5.2 <i>Saran</i>	75

DAFTAR PUSTAKA

LAMPIRAN A (*LISTING PROGRAM*)

LAMPIRAN B (*TABLE T STUDENT*)

**LAMPIRAN C (SURAT KETERANGAN HASIL
PENGAMBILAN DATA MONITORING
GAS KARBONMONOKSIDA)**

LAMPIRAN D (*DATASHEET* SENSOR GAS MQ 7)

DAFTAR GAMBAR

	Hal
Gambar 2.1	Sensor Gas MQ-79
Gambar 2.2	Karakteristik Sensitivitas Sensor MQ-711
Gambar 2.3	STM32F407 <i>Discovery</i>12
Gambar 2.4	LCD 16X414
Gambar 2.5	RTC DS 32312215
Gambar 2.6	<i>Module SD Card</i>16
Gambar 2.7	<i>Modem Wavecom Fastrack M1206B Serial</i>17
Gambar 2.8	Arsitektur <i>Modem Wavecom Fastrack M1306B Serial</i>18
Gambar 2.9	<i>Pinout RS232</i>19
Gambar 2.10	<i>Exhaust Fan DC 12V</i>21
Gambar 2.11	<i>Power supply Switching 12VDC</i>22
Gambar 2.12	MCB Shukaku 220 V 2A.....23
Gambar 2.13	IC 780524
Gambar 2.14	Logo <i>ChibiOs</i>25
Gambar 2.15	Tampilan Qt Creator26
Gambar 2.16	Diagram Alir Penentuan Nilai Ketidakpastian Baku dari Data Tipe A dan B....27
Gambar 3.1	Flowchart Pengerjaan Tugas Akhir33
Gambar 3.2	Diagram Blok Sistem Pengukuran.....34
Gambar 3.3	Desain Alat pada <i>Software Sketch Up</i>36
Gambar 3.4	Skematik <i>CO Gas Sensor Module</i>37
Gambar 3.5	Modul sensor MQ-7.....37
Gambar 3.6	Skematik IC LM 39338
Gambar 3.7	Konfigurasi pin STM32F4 <i>Discovery</i>39
Gambar 3.8	Skematik <i>Shield Board STM32F4 Discovery</i>41
Gambar 3.9	Rangkaian Skematik LCD pada STM32F4 <i>Discovery</i>41
Gambar 3.10	Penempatan LCD untuk <i>Display</i>42
Gambar 3.11	Skematik Real Time Clock (RTC) pada STM32F4 <i>Discovery</i>42
Gambar 3.12	Modul RTC DS3231.....43

Gambar 3.13	Skematik <i>Modul SD Card</i> pada STM32F4 <i>Discovery</i>	43
Gambar 3.14	<i>Modul SD Card</i>	44
Gambar 3.15	Konfigurasi pin modem wavecom.....	45
Gambar 3.16	<i>Modem Wavecom M1306B</i>	46
Gambar 3.17	Blok Diagram <i>Power Supply 12VDC</i>	46
Gambar 3.18	<i>Power Supply</i>	47
Gambar 3.19	Rangkaian Skematik Mikrokontroler STM32F4, <i>sensor MQ-7</i> , <i>SD Card</i> , <i>RTC</i> , <i>Uart (SMS Gateway)</i> dan <i>LCD 16x4</i>	48
Gambar 3.20	<i>Create New Project</i>	49
Gambar 3.21	<i>Project Qt Creator</i>	49
Gambar 3.22	<i>Import Existing Project Name and Location</i>	50
Gambar 3.23	<i>Import Existing Project File Selection</i>	50
Gambar 3.24	<i>Import Existing Project Management</i>	51
Gambar 3.25	Tampilan Awal Program.....	51
Gambar 3.26	<i>Program Project .files</i>	52
Gambar 3.27	<i>Program Project .includes</i>	52
Gambar 3.28	<i>Class pada Project</i>	53
Gambar 3.29	<i>Build Project Qt Creator</i>	53
Gambar 3.30	<i>Make all Project di Notepad++</i>	54
Gambar 3.31	<i>Download Project di ST-LINK V2</i>	54
Gambar 4.1	Alur Sistem <i>Monitoring Gas</i>	55
Gambar 4.2	Alat <i>Monitoring Gas</i>	56
Gambar 4.3	Pengujian data waktu <i>RTC</i> pada <i>Hyperterminal</i>	57
Gambar 4.4	Pengujian <i>SD Card</i>	58
Gambar 4.5	<i>Screenshoot AT Command</i> pada <i>Hyperterminal</i>	59
Gambar 4.6	<i>Screenshoot SMS</i> yang diterima pada <i>HP</i>	59
Gambar 4.7	<i>Screenshoot SMS</i> yang diterima pada <i>HP</i> melebihi Nilai Ambang Batas.....	60
Gambar 4.8	Grafik Uji Sensor dengan 2 keadaan	61
Gambar 4.9	Grafik Pembacaan Standar dan Pembacaan	

	Alat	65
Gambar 4.10	Grafik Pembacaan Standar dan Pembacaan Alat dengan tipe <i>scatter</i>	65
Gambar 4.11	Grafik Pembacaan Standar dan Pembacaan Alat dengan persamaan matematik	66

DAFTAR TABEL

	Hal
Tabel 2.1	Baku mutu faktor fisika dan dan faktor kimia6
Tabel 2.2	Efek Paparan Gas CO8
Tabel 2.3	Spesifikasi sensor MQ-79
Tabel 2.4	Pin Konfigurasi LCD 16x414
Tabel 2.5	Pin RS23219
Tabel 3.1	Konfigurasi pin STM32F4 <i>Discovery</i>39
Tabel 4.1	Uji Sensor Dalam 2 Kondisi61
Tabel 4.2	Pengambilan Data Monitoring62
Tabel 4.3	Tabel Konversi Data ADC ke PPM64
Tabel 4.4	Data Spesifikasi Alat67
Tabel 4.5	Data Kalibrasi69

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi memberikan dampak yang besar bagi kelangsung hidup manusia terutama masalah lingkungan. Pencemaran udara telah menjadi salah satu topik yang sering kali menjadi bahan permasalahan lingkungan. Polusi udara merupakan masalah lingkungan global yang terjadi di seluruh dunia. Berdasarkan data dari *World Health Organization* (WHO), polusi udara menyebabkan kematian prematur mencapai 2 juta jiwa pertahun [1].

Indonesia menjadi salah satu negara dengan tingkat pencemaran udara yang tinggi, kurang lebih 70% terjadi gangguan kesehatan di daerah dengan pencemaran udara yang tinggi seperti Jakarta, Medan, Batam dan Solo [2].

Polusi udara bersumber pada proses alami dan aktivitas manusia, bergerak maupun tidak bergerak. Kebanyakan masalah pencemaran udara di perkotaan bersumber dari penggunaan bahan bakar fosil dan kegiatan perindustrian. Asap kendaraan bermotor memiliki peranan penting sebagai sumber polusi udaraterbesar mencapai 60-70%, dibanding dengan industri yang hanya berkisar antara 10-15% [3]. Sedangkan sisanya berasal dari rumah tangga, pembakaran sampah, kebakaran hutan atau ladang dan lain-lain. Hal ini diakibatkan oleh peningkatan kepemilikan kendaraan bermotor yang bertambah banyak tiap tahunnya. Sebagian besar kendaraan bermotor tersebut menghasilkan emisi gas buang yang buruk, dikarenakan perawatan mesin yang kurang memadai ataupun dari penggunaan bahan bakar dengan kualitas kurang baik [4].

Beberapa macam polutan yang dihasilkan dari emisi gas buang yang buruk antara lain gas Karbon Monoksida (CO), Hidrokarbon (HC), Nitrogen Oksida (NO), Sulfur Oksida (SO₂) dan Timbal (Pb) yang sering disebut sebagai polutan primer. Salah satu polutan udara yang berbahaya dan yang sangat dominan jumlahnya adalah gas CO yang dihasilkan dari proses pembakaran bahan bakar dan udara motor bensin yang tidak sempurna [5].

Karbon monoksida masuk ke atmosfer melalui gas buang dan akan cepat teroksidasi membentuk karbondioksida. Karbon monoksida (CO) berbahaya karena tingkat toksisitasnya yang tinggi terhadap manusia dan hewan. Waktu tinggal karbon monoksida (CO) di atmosfer antara 1 sampai 2 bulan. Waktu paruh CO terikat dalam darah kira-kira 250 menit [6].

Kota Surabaya menjadi salah satu kota yang memiliki mobilitas tinggi dalam hal penggunaan kendaraan bermotor. Menurut Kepala Badan Lingkungan Hidup (BLH) Provinsi Jawa Timur, Dewi J Putriatni, menyebutkan, Surabaya menduduki peringkat ketiga setelah Bangkok dan Jakarta sebagai kota di kawasan Asia yang yang polusi udaranya buruk. Dalam memantau kondisi tingkat polusi udara yang ada di wilayah Surabaya, Pemerintah Kota Surabaya juga telah memiliki sarana dan prasarana berupa ISPU (Indeks Standar Pencemaran Udara). Dari 6 unit ISPU yang beroperasi hanya ada 2 unit yang masih aktif yakni di jalur MERR dan kawasan Hotel Sahid [7].

Oleh karena itu, dalam tugas akhir ini akan dirancang dan dibangun sebuah sistem monitoring konsentrasi gas karbon monoksida (CO) yang berfungsi untuk mengetahui konsentrasi atau konsentrasi adanya emisi gas buang karbon monoksida (CO) pada suatu area tertentu, dimana penulis mengkhususkan penggunaan hasil data monitoring tersebut kedalam penyimpanan data logger dan mengirimkan informasi tersebut melalui *sms gateway*. Penulis mengangkat topik tersebut dengan judul **“Rancang Bangun Sistem Monitoring Gas Karbon Monoksida (CO) Sebagai Emisi Gas Buang Menggunakan Sensor Gas MQ-7 Berbasis Mikrokontroller STM32F4 Discovery”**

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

- a. Bagaimana merancang suatu sistem monitoring konsentrasi gas karbon monoksida (CO) yang dihasilkan

dari emisi gas buang menggunakan sensor gas MQ 7 berbasis mikrokontroler STM32F4 *Discovery*?

- b. Bagaimana hasil pengukuran dari alat yang telah dibuat?
- c. Bagaimana penggunaan hasil data monitoring konsentrasi gas karbon monoksida (CO) yang dihasilkan dari emisi gas buang?

1.3 Tujuan

Tujuan yang dicapai dalam tugas akhir ini adalah sebagai berikut:

- a. Membangun dan merancang suatu sistem monitoring konsentrasi gas karbon monoksida (CO) yang dihasilkan dari emisi gas buang menggunakan sensor gas MQ 7 berbasis mikrokontroler STM32F4 *Discovery*
- b. Mengetahui hasil pengukuran dari alat yang telah dibuat.
- c. Mengetahui penggunaan hasil data monitoring konsentrasi gas karbon monoksida (CO) yang dihasilkan dari emisi gas buang

1.4 Batasan Masalah

Perlu diberikan beberapa batasan masalah agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan masalah dari sistem yang dirancang ini adalah sebagai berikut:

- a. Perancangan sistem monitoring konsentrasi gas karbon monoksida (CO) menggunakan sebuah mikrokontroler STM32F4 *Discovery*.
- b. Perancangan sistem monitoring konsentrasi gas karbon monoksida (CO) menggunakan sensor MQ-7.
- c. Perancangan sistem monitoring konsentrasi gas karbon monoksida (CO) menggunakan sebuah *display* LCD ukuran 16x4 karakter.
- d. Pengujian sistem dari rancang bangun yang telah dibuat dengan menguji performansi alat, baik keakuratan dan keoptimalan alat.

- e. Penyusunan hasil teori dari pembuatan *hardware*, analisa data dan kesimpulan dari data dan sistem yang ada.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai sistem monitoring konsentrasi gas karbon monoksida (CO) menggunakan mikrokontroler STM32F4 *Discovery* dan dapat dijadikan sebagai perancangan alat monitoring ISPU (Indeks Standar Pencemaran Udara) masa depan.

1.6 Sistematika Laporan

Sistematika laporan yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini membahas mengenaiS latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, dan sistematika penulisan dalam tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Pada bab ini membahas mengenai teori-teori penunjang yang diperlukan dalam merealisasikan tugas akhir yaitu berupa teori tentang gas karbon monoksida, emisi gas karbon monoksida, nilai ambang batas, tanda dan gejala keracunan gas karbon monoksida (CO) dan perangkat-perangkat yang digunakan dalam pembuatan tugas akhir ini.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini diuraikan tentang penjelasan mengenai perancangan dan pembuatan alat.

BAB IV PENGUJIAN DAN ANALISIS DATA

Pada bab ini memuat tentang hasil pengujian dari perangkat yang dibuat beserta pembahasannya.

BAB V PENUTUP

Pada bab ini memuat tentang kesimpulan dan saran dari pembuatan tugas akhir ini.

BAB II TINJAUAN PUSTAKA

2.1 Gas Karbon Monoksida (CO)

Karbon monoksida (CO) adalah suatu gas yang tidak berwarna, tidak berbau, tidak berasa, mudah terbakar, tidak mengiritasi namun sangat beracun. Dari sifat-sifat tersebut karbon monoksida dikenal sebagai “*silent killer*”. Untuk mengukur konsentrasi CO tersebut, digunakan *Gas Analyzer* dengan satuan persen *volume*. Satuan konsentrasi CO di udara adalah ppm atau parts per million. Dimana 1 ppm setara dengan $10^{-4}\%$ [8].

Karbon monoksida (CO) merupakan salah satu bahan toksik yang sangat berbahaya jika terhirup oleh manusia. Gas CO dapat terbentuk secara alamiah, tetapi sumber utamanya adalah dari kegiatan manusia. Korban monoksida yang berasal dari alam termasuk dari lautan, oksidasi metal di atmosfer, pegunungan, kebakaran hutan dan badai listrik alam. Gas CO dapat ditemukan dari hasil pembakaran yang tidak sempurna dari karbon dan bahan-bahan organik yang mengandung karbon. Sumber terpenting gas CO adalah asap kendaraan bermotor yang menggunakan bensin sebagai bahan bakar, karena campuran bahan yang terbakar mengandung bahan bakar lebih banyak daripada udara, sehingga gas yang dikeluarkan mengandung 3-7% CO, sebaliknya motor diesel dengan *compression ignition* mengeluarkan sangat sedikit CO, kecuali bila motor berfungsi tidak sempurna sehingga banyak mengeluarkan asap hitam yang mengandung CO [9].

Berdasarkan estimasi, jumlah CO dari sumber buatan diperkirakan mendekati 60 juta Ton per tahun. Separuh dari jumlah ini berasal dari kendaraan bermotor yang menggunakan bahan bakar bensin dan sepertiganya berasal dari sumber tidak bergerak seperti pembakaran batubara dan minyak dari industri dan pembakaran sampah domestik. Gas CO ini tidak hanya dihasilkan oleh kendaraan bermotor saja tetapi juga dihasilkan dari

asap rokok, asap pabrik, alat pemanas, dan peralatan yang menggunakan bahan api berasaskan karbon [10].

Gas CO yang keluar dari knalpot akan berada di udara *ambient*, jika terhirup oleh manusia maka molekul tersebut akan masuk kedalam saluran pernapasan terus masuk ke dalam paru-paru dan kemudian akan menempel pada haemoglobin darah membentuk karboksi *hemoglobin* (COHb).[10] Semakin tinggi konsentrasi CO yang terhirup oleh manusia maka semakin fatal resiko yang diterima oleh manusia tersebut, bahkan dapat menyebabkan kematian [11].

2.2 Emisi Gas Buang Karbon Monoksida (CO)

Menurut PP No.29 tahun 1986, pencemaran udara dapat juga diartikan berubahnya tatanan udara oleh kegiatan manusia atau oleh proses alam sehingga kualitas udara menjadi kurang atau tidak dapat berfungsi lagi sesuai dengan kegunaannya. Emisi ini berasal dari sumber polusi, yang utama berasal dari transportasi, dimana hampir 60% dari polutan yang dihasilkan terdiri dari karbon monoksida (CO) dan sekitar 15% terdiri dari hidrokarbon (HC).

Menurut laporan WHO (1992) dinyatakan paling tidak 90% dari karbon monoksida (CO) di udara perkotaan berasal dari emisi kendaraan bermotor. Selain itu asap rokok juga mengandung karbon monoksida (CO), sehingga para perokok dapat memajan dirinya sendiri dari asap rokok yang sedang dihisapnya [12].

2.3 Nilai Ambang Batas

Nilai ambang batas adalah konsentrasi dari zat ,uap , atau gas dalam udara yang dihirup selama 8 jam per hari dan 40 jam selama satu minggu, tanpa menimbulkan gangguan kesehatan yang sangat berarti. berdasarkan PER.13/MEN/X/2011 tentang faktor fisika dan faktor kimia di tempat kerja dapat dilihat pada tabel 2.1 di bawah ini:

Tabel 2.1 Baku mutu faktor fisika dan dan faktor kimia

No	Parameter	Nilai Ambang Batas (NAB) PPM
1	Karbon dioksida	5000
2	Karbon disulfida	10

Tabel 2.1 (Lanjutan)

3	Karbon monoksida	25
4	Nitrit oksida	25
5	Gasolin	300

Nilai ambang batas CO di tempat kerja yaitu 25 ppm. Keadaan normal konsentrasi karbon monoksida di dalam darah berkisar antara 0,2%-1,0%, dan rata-rata sekitar 5% COHb [13].

2.4 Tanda dan Gejala Keracunan Karbon Monoksida (CO)

Kontak antara manusia dengan karbon monoksida pada konsentrasi tinggi dapat menyebabkan kematian. tetapi ternyata kontak dengan karbon monoksida pada konsentrasi relatif rendah (100 ppm atau kurang) juga dapat mengganggu kesehatan. Karbon monoksida memiliki kemampuan berikatan dengan hemoglobbin yang mengakibatkan penurunan kapasitas oksigen yang dibawa dalam darah, merusak pelepasan oksigen ke jaringan dan mempengaruhi proses oksidasi intraselular yang mengakibatkan hipoksia jaringan [14].

Keracunan gas karbon monoksida bisa bersifat kronis. Gejala yang ditimbulkan dari efek keracunan ini adalah sakit kepala, pening, berkunang-kunang, lemah, ngilu persendian, mual, dan muntah-muntah, sesak nafas terutama pada saat berolah raga, dan gangguan penglihatan. Keracunan kronis akibat karbon monoksida ini sulit di diagnosis terutama oleh dokter atau tenaga medis yang belum berpengalaman. Kadang dilihat dari gejala di diagnosis sebagai infeksi penyakit viral atau bakterial pada paru atau gastro intestinal atau sindrom lainnya. Gejala yang mirip tersebut sering terjadi pada satu individu dan gejala tersebut dapat menurun kemudian hilang dengan sendirinya pada saat polusi lingkungan tersebut telah menurun atau hilang.

Di udara menurut WHO paparan karbon monoksida, 87 ppm selama 15 menit, 52 ppm selama 30 menit, 26 ppm selama 1 jam, 9 ppm selama 8 jam. Konsentrasi gas karbon monoksida (CO) di udara secara langsung akan mempengaruhi konsentrasi karboksihemoglobin (COHb). Bila konsentrasi gas CO di udara tetap maka konsentrasi COHb di dalam

darah akan mencapai keseimbangan tertentu dan akan tetap bertahan selama tidak ada perubahan pada konsentrasi CO di udara. Dalam keadaan normal sebenarnya darah sudah mengandung COHb sebanyak 0,5%, berasal dari proses metabolisme di dalam tubuh, terutama merupakan hasil pemecahan heme komponen hemoglobin dalam darah itu sendiri, di tambah lagi konsentrasi CO yang terdapat di udara dalam konsentrasi rendah. Hubungan antara konsentrasi COHb dalam darah dengan konsentrasi di udara (<100 ppm) adalah sebagai berikut:

$$\% \text{ COHb dalam darah} = 0,16 (\text{konsentrasi CO di udara}) + 0,5 \dots \dots \dots (2.1)$$

Dibawah ini merupakan tabel persamaan hubungan konsentrasi COHb dalam darah dengan konsentrasi CO di udara:

Tabel 2.2 Efek Paparan Gas CO^[15]

Konsentrasi Rata-Rata 8 jam (ppm)	Konsentrasi COHb dalam darah (%)	Gejala
25-50	2,5-5	Tidak ada gejala
50-100	5-10	Aliran darah meningkat, sakit kepala ringan
100-250	10-20	Tegang daerah dahi, sakit kepala, penglihatan agak terganggu
250-450	20-30	Sakit kepala sedang, berdenyut-denyut, dahi (<i>throbbing temple</i>), wajah merah dan mual
450-650	30-40	Sakit kepala berat, vertigo, mual, muntah, lemas, mudah terganggu, pingsan pada saat bekerja
650-1000	40-50	Mudah pingsan dan jatuh
1000-1500	50-60	Koma, hipotensi, kadang disertai kejang, pernafasan <i>Cheyne- Stokes</i>
1500-2500	60-70	Koma dengan kejang, penekanan pernafasan dan fungsi jantung, mungkin terjadi kematian
2500-4000	70-80	Denyut nadi lemah, pernafasan lambat, gagal hemodinamik, kematian

2.5 Sensor Gas MQ-7

Sensor MQ-7 merupakan sensor gas karbon monoksida (CO) yang berfungsi untuk mengetahui konsentrasi gas karbon monoksida (CO). Dimana sensor ini salah satunya dipakai dalam memantau gas karbon monoksida (CO). Sensor ini memiliki range pembacaan konsentrasi sebesar 10-500 ppm. Sensor ini memiliki sensitivitas tinggi dan waktu respon yang cepat [16].



Gambar 2.1 Sensor Gas MQ-7

Keluaran yang dihasilkan oleh sensor ini adalah berupa sinyal analog. Sensor ini juga membutuhkan tegangan direct current (DC) sebesar 5V. Pada sensor ini terdapat nilai resistansi sensor (R_s) yang dapat berubah bila terkena gas dan juga sebuah pemanas yang digunakan sebagai pembersihan ruangan sensor dari kontaminasi udara luar. Sensor ini memerlukan rangkaian sederhana serta memerlukan tegangan pemanas (*power heater*) sebesar 5V, resistansi beban (*load resistance*), dan *output* sensor dihubungkan ke *analog digital converter (ADC)*, sehingga keluaran dapat ditampilkan dalam bentuk sinyal *digital*. Maka nilai *digital* yang berupa *output* sensor ini dapat ditampilkan pada sebuah *liquid crystal display (LCD)* atau alat penampil lainnya. Pada tabel 2.3 dijelaskan spesifikasi yang terdapat pada sensor gas MQ-7.

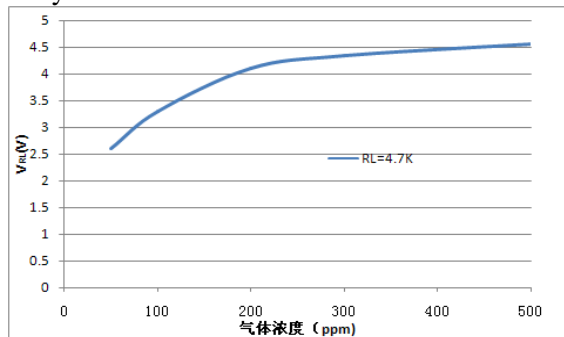
Tabel 2.3 Spesifikasi sensor MQ-7

<i>Model</i>	<i>MQ-7</i>
<i>SensoType</i>	<i>Semiconductor</i>
<i>Standard Encapsulation</i>	<i>Plastic cap</i>
<i>Target</i>	<i>carbon monoxide</i>

Tabel 2.3 (Lanjutan)

<i>Detection range</i>			<i>10~500ppm CO</i>
<i>Standard Circuit Conditions</i>	<i>Loop Voltage</i>	<i>V_c</i>	$\leq 10V$ DC
	<i>Heater Voltage</i>	<i>V_H</i>	5.0V \pm 0.1V AC or DC (High tem.) 1.5V \pm 0.1V AC or DC (Low tem.)
	<i>Heater Time</i>	<i>T_L</i>	60 S \pm 1S (High Tem), 90 S \pm 1S (Low tem.)
	<i>Load Resistance</i>	<i>R_L</i>	Adjustable
<i>Sensor character under standard test conditions</i>	<i>Heater Resistance</i>	<i>R_H</i>	29 Ω \pm 3 Ω (room tem.)
	<i>Heater consumption</i>	<i>P_H</i>	$\leq 900mW$
	<i>Sensitivity</i>	<i>S</i>	$R_s(\text{in air})/R_s(\text{in})$
	<i>Output Voltage</i>	<i>V_s</i>	2.5V~4.3V (in
	<i>Concentration Slope</i>	<i>A</i>	$\leq 0.6(R300ppm/R50ppm CO)$
<i>Standard test conditions</i>	<i>Tem. Humidity</i>		20 °C \pm 2 °C; 55% \pm 5 %RH
<i>Standard test conditions</i>	<i>Standard test circuit</i>		V _c :5.0V \pm 0.1V; V _H (High tem.): 5.0V \pm 0.1V; V _H (Low tem.): 1.5V \pm 0.1V
	<i>Preheat time</i>		Over 48 hours

Pada gambar dibawah merupakan grafik karakteristik sensitivitas nilai konsentrasi gas dari sensor MQ-7. Nilai konsentrasi gas (ppm) dari sensor ini meningkat jika nilai resistansi (R_s/R_o) menurun atau lebih kecil dari nilai konsentrasinya.



Gambar 2.2 Karakteristik Sensitivitas Sensor MQ-7

2.6 Mikrokontroler

Sebuah komputer mikro memiliki tiga komponen utama yaitu, unit pengolah pusat *Central Processing Unit (CPU)*, memori dan sistem *input/output (I/O)* untuk dihubungkan dengan perangkat luar. *Central Processing Unit (CPU)*, yang mengatur sistem kerja komputer mikro, dibangun oleh sebuah mikroprosesor. Memori terdiri atas EEPROM untuk menyimpan program dan RAM untuk menyimpan data. Sistem *I/O* bisa dihubungkan dengan perangkat luar misalnya sebuah *keyboard* dan sebuah *monitor*, bergantung pada aplikasinya. Apabila *Central Processing Unit (CPU)*, memori dan sistem *I/O* dibuat dalam sebuah *chip* semikonduktor, maka dinamakan mikrokontroler [17].

2.7 Mikrokontroler STM32F4 Discovery

Arsitektur ARM merupakan arsitektur *processor* 32-bit RISC yang dikembangkan oleh ARM Limited dikenal sebagai *Advanced RISC Machine* dimana sebelumnya dikenal sebagai *Acorn RISC Machine*. Pada awalnya merupakan *prosesor desktop*

yang sekarang didominasi oleh keluarga x86. Namun desain yang sederhana membuat prosesor ARM cocok untuk aplikasi berdaya rendah. Hal ini membuat prosesor ARM mendominasi pasar *mobile electronic* dan embedded sistem dimana membutuhkan daya dan harga yang rendah. Karena penggunaan AT MEGA dari ATMEL sudah mulai ditinggalkan dengan sudah terlalu banyak aplikasi dengan AT MEGA maka harus berkembang dengan ARM yang harganya lebih murah dengan teknologi yang lebih canggih. STMicroelectronics adalah salah satu *vendor ARM* yang memiliki market share terbesar. Harga STM32 Discovery Board yang cukup ekonomis serta memiliki kelengkapan yang *excellent*, lebih dari sekedar minimum sistem. Bahkan secara keseluruhan, lebih murah *development board* berbasis mikrokontroler 8-bit. STM32 Discovery Board dapat dijadikan media pembelajaran platform 32-bit. ARM Cortex-M yang mumpuni. Di dalamnya sudah dilengkapi dengan ST- LINK/V2 untuk *programming* dan *debugging* melalui koneksi USB. STM32 Discovery Board juga dapat digunakan untuk membangun aplikasi dengan tingkat kompleksitas algoritma yang cukup tinggi, karena dicatu prosesor kelas 32-bit berkinerja tinggi [18].



Gambar 2.3 STM32F407 Discovery

Fitur utama

- a. Mikrokontroler STM32F407VGT6 menampilkan 32-bit ARM® Cortex® -M4 dengan FPU inti, 1-Mbyte memori Flash, RAM 192-Kbyte dalam paket LQFP100
- b. On-board ST-LINK / V2 pada STM32F4 DISCOVERY (referensi tua) atau ST-LINK / V2-A pada STM32F407G-DISC1 (kode orde baru) USB ST-LINK dengan kemampuan

- re-pencacahan dan tiga antarmuka yang berbeda yakni debug port, virtual port Com (dengan kode orde baru saja), dan mass storage (dengan kode orde baru saja)
- c. Dewan *power supply*: melalui bus USB atau dari tegangan suplai 5 Volt eksternal
 - d. Eksternal *power supply* aplikasi: 3 V dan 5 V
 - e. LIS302DL atau LIS3DSH ST MEMS *accelerometer 3-axis*
 - f. MP45DT02 ST-MEMS sensor audio yang *omni-directional microphone digital*
 - g. DAC audio yang CS43L22 dengan kelas yang terintegrasi sopir D *speaker*
 - h. Delapan LED, yakni:
 - LD1 (merah / hijau) untuk komunikasi USB
 - LD2 (merah) untuk 3,3 V daya pada
 - Empat LED pengguna, LD3 (oranye), LD4 (hijau), LD5 (merah) dan LD6 (biru)
 - 2 USB OTG LED LD7 (hijau) VBUS dan LD8 (merah) over-saat ini
 - i. Dua *push-tombol* (pengguna dan reset)
 - j. *USB OTG FS* dengan konektor *micro-AB extension header* untuk semua LQFP100 I / Os untuk koneksi cepat ke papan prototyping dan mudah menyelidiki perangkat lunak bebas yang komprehensif termasuk berbagai contoh, bagian dari STM32CubeF4 paket atau STSW-STM32068 menggunakan standar perpustakaan warisan [19].

2.8 LCD 16X4

Liquid crystal display (LCD) merupakan sejenis *crystal* yang akan berpendar jika diberi tegangan tertentu, sehingga perpendaran tersebut dapat diatur untuk membentuk karakter, angka, huruf dan lain sebagainya. LCD dapat diprogram agar sesuai dengan aplikasi yang telah dirancang. LCD pada prinsipnya sama dengan penampil dot matrik. Terdapat dua jenis LCD yaitu LCD teks dan LCD grafik. LCD teks adalah jenis LCD yang digunakan untuk menampilkan teks atau angka dalam kode

ASCII. LCD teks yang ada dalam teks dibagi dalam sel, dimana setiap selnya hanya dapat menampilkan karakter ASCII. Sedangkan *graphic* LCD dapat menampilkan gambar.

Liquid crystal display (LCD) yang digunakan dalam penelitian tugas akhir ini adalah menggunakan *liquid crystal display (LCD) teks* ukuran 16x4, dimana LCD ini mempunyai 4 baris dan setiap barisnya dapat menampilkan hingga 16 karakter. Pemrograman LCD karakter pada CVAVR relatif lebih mudah dibandingkan jika menggunakan *software* standar C, seperti WinAVR [20].



Gambar 2.4 LCD 16X4

Tabel 2.4 Pin Konfigurasi LCD 16x4

Fungsi	PIN	Deskripsi	Data Logika	Keterangan
Ground	1	VSS (Ground)	-	0 V
Tegangan Sumber	2	VDD (VCC)	-	+5 V
Kontras	3	VEE	-	0-Vdd
Kendali LCD	4	RS	0/1	
Data	5	R/W	0/1	
	6	E	0/1/floating	
	7	D0		Bit 0
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7

Tabel 2.4 (Lanjutan)

Anoda	15	<i>Backlight, lampu background</i>	-	
Katoda	16	<i>Backlight, lampu background</i>	-	

2.9 RTC DS3231

DS3231 adalah I2C sangat akurat *real-time clock* (RTC) dengan *temperature compensated* terintegrasi osilator kristal (TCXO) dan kristal. Perangkat menggabungkan input baterai, dan memelihara ketepatan waktu yang akurat ketika listrik utama ke perangkat terganggu. RTC mempertahankan detik, menit, jam, hari, tanggal, bulan, dan informasi tahun. Tanggal pada akhir bulan secara otomatis disesuaikan dengan bulan dengan lebih sedikit dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi baik dalam 24 jam atau 12 jam dengan format AM indikator / PM. Diprogram dua waktu-of-hari alarm dan *output* gelombang persegi programmable yang disediakan. Alamat dan data yang ditransfer serial melalui dua arah bus I2C.

**Gambar 2.5** RTC DS 3231

Referensi tegangan kompensasi suhu presisi dan sirkuit komparator memonitor status VCC yang berfungsi untuk mendeteksi kegagalan daya, memberikan keluaran reset, dan digunakan secara otomatis beralih ke persediaan cadangan bila diperlukan. Selain itu, pin RST dipantau sebagai masukan tombol tekan untuk menghasilkan reset secara eksternal [21].

2.10 Micro SD Sield Modul

Modul (*MicroSD Card Adapter*) adalah modul pembaca kartu *Micro SD*, dan antarmuka *SPI* melalui driver sistem file, sistem mikrokontroler untuk melengkapi kartu *MicroSD* membaca dan menulis file. Pengguna *STM32F4* bisa langsung menggunakan *ChibiOs* hadir dengan kartu *SD* untuk melengkapi inisialisasi kartu perpustakaan dan baca tulis. Terdapat enam pin (*GND*, *VCC*, *MISO*, *MOSI*, *SCK*, *CS*), *GND* ke *ground*, *VCC* adalah *power supply*, *MISO*, *MOSI*, *SCK* adalah bus *SPI*, *CS* adalah *chip select signal* pin [22].



Gambar 2.6 *Module SD Card*

2.11 Data Logger

Data logger adalah suatu alat elektronik yang berfungsi mencatat data dari waktu ke waktu secara *continue*. Beberapa *data logger* menggunakan personal komputer dan *software* sebagai tempat menyimpan data dan menganalisis data. Data yang disimpan di *harddisk* dapat diakses kapanpun kita inginkan. Hal ini termasuk beberapa perangkat akuisisi data seperti *plug-in board* atau sistem komunikasi serial yang menggunakan komputer sebagai sistem penyimpanan data real time.

Data logger berbasis PC (*PC-based data logger*) menggunakan komputer, biasanya PC, untuk mengumpulkan data melalui sensor dalam rangka menganalisis dan menampilkan hasilnya. *Data logger* berbasis PC (*PC-based data logger*) menggunakan komputer, biasanya PC, untuk mengumpulkan data melalui sensor dalam rangka menganalisis dan menampilkan hasilnya.

Salah satu keuntungan menggunakan *data logger* adalah kemampuannya secara otomatis mengumpulkan data setiap 24

jam. Setelah diaktifkan, *data logger* digunakan dan ditinggalkan untuk mengukur dan merekam informasi selama periode pemantauan. Hal ini memungkinkan untuk mendapatkan gambaran yang komprehensif tentang kondisi lingkungan yang dipantau [23].

2.12 Modem GSM Wavecom Fastrack M1206B

Wavecom M1306B adalah GSM/GPRS modem yang siap digunakan sebagai modem untuk suara, data, fax dan SMS. Kelas ini juga mendukung 10 tingkat kecepatan transfer data. Wavecom M1306B TCP/IP dengan mudah dikendalikan dengan menggunakan perintah AT untuk semua jenis operasi karena mendukung fasilitas koneksi RS232 dan juga fasilitas TCP/IP *stacked*. Dapat dengan cepat terhubung ke *port* serial komputer *desktop* atau *notebook*. *casing* logam Wavecom M1206B TCP/IP menjadi solusi yang tepat untuk aplikasi berat seperti telemetri atau *Wireless Local Loop* (PLN metering & Telepon Umum). Ukurannya sangat kecil memudahkan dalam peletakkan di berbagai macam area, *indoor/outdoor*. Cocok sekali digunakan pada aplikasi: Server Pulsa yang menghendaki kemampuan optimal dan usia pakai panjang, telemetri, SMS *gateway/broadcast* yang handal, PPOB PLN, ATM, *Payment Point Systems*, Metering Listrik. Modem GSM Serial Wavecom Fastrack M1206B memiliki dua *type* konektor yaitu *serial* dan *USB* [24].

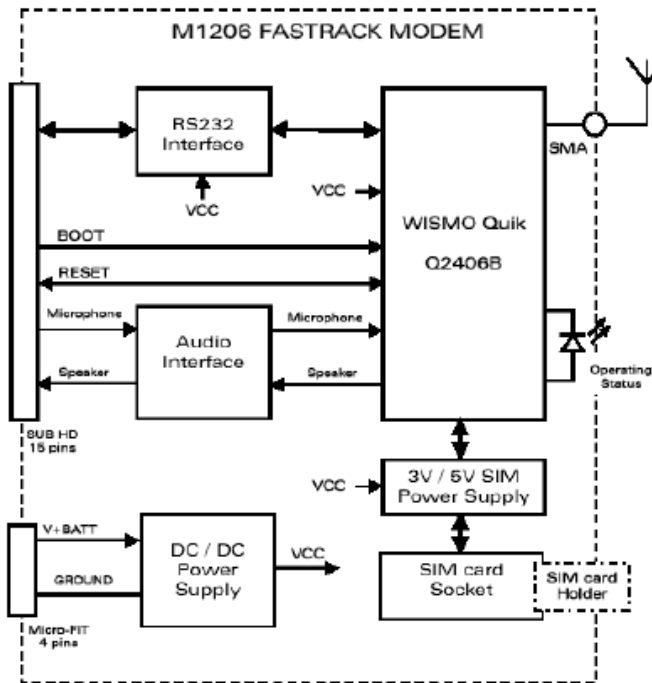


Gambar 2.7 Modem Wavecom Fastrack M1206B Serial
Spesifikasi modem WAVECOM FASTRACK M1206B:

- a. *Dual-band* GSM 900/1800MHZ & GPRS *Class* 10

- b. *GSM Dual Band antenna*
- c. *Power supply with 4 pin connector (untuk serial)*
- d. *Standard USB 2.0 interface (untuk USB)*
- e. *Input Voltage : 5V-32V*
- f. *Maximum transmitting speed 253KBps*
- g. *Support AT-Command*
- h. *Dimensi : 74×54×25mm*

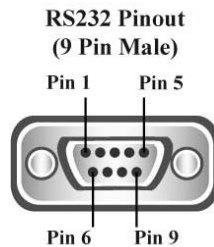
Berikut ini arsitektur dari Modem GSM Wavecom *Fastrack M1206B*:



Gambar 2.8 Arsitektur Modem Wavecom *Fastrack M1306B*
Serial

2.13 RS232

RS232 adalah *standard* komunikasi *serial* yang digunakan untuk koneksi periperal ke periperal. Biasa juga disebut dengan jalur I/O (*input/output*). Standar ini menggunakan beberapa piranti dalam implementasinya. Paling umum yang dipakai adalah *plug/ konektor* DB9 atau DB25. *Port Serial* RS232 juga mempunyai fungsi yaitu untuk menghubungkan / koneksi dari perangkat yang satu dengan perangkat yang lain, atau peralatan *standart* yang menyangkut komunikasi data antara komputer dengan alat-alat pelengkap komputer.



Gambar 2.9 Pinout RS232

Komunikasi data secara *sesial* dilakukan dengan metode untuk mengirimkan data dari sebuah pengirim secara bit per bit dengan kecepatan tertentu (bit per detik/bps), dan penmgiriman dilakukan melalui jalur satu kawat (Tx) dan diterima oleh sebuah penerima (Rx) dalam waktu tertentu. Sebuah rangkaian RS232 yang dapat mengubah jalur paralel menjadi jalur serial ditunjuk oleh sebuah IC tipe 6850, yaitu sebuah rangkaian yang dikenal dengan istilah *Asynchronous Communications Interface Adapter (UART)* [25]. Dibawah ini merupakan keterangan pin RS232 terdapat pada tabel 2.5

Tabel 2.5 Pin RS232

No.	Pin	Keterangan
1	Pin 1	N/C
2	Pin 2	RXD
3	Pin 3	TXD

Tabel 2.5 (Lanjutan)

4	Pin 4	DTR
5	Pin 5	GND
6	Pin 6	DSR
7	Pin 7	RTS
8	Pin 8	CTS
9	Pin 9	<i>Power Input</i>

Fungsi pin berdasarkan tabel 2.5 dapat dijelaskan sebagai berikut:

- a. *Signal Ground (SG)* berfungsi untuk memberikan masa (*ground*) pada setiap sinyal secara bersama (*common signal ground*).
- b. *Transmit Data (TX)* berfungsi sebagai saluran keluarnya data dari UART atau sebagai pengirim data ke *device* secara *serial*.
- c. *Receive Data (RX)* berfungsi sebagai saluran masuknya data ke UART atau sebagai penerima data dari devais secara serial.
- d. *Data Terminal Ready (DTR)* berfungsi sebagai pemberi informasi status ke devais terkoneksi bahwa UART telah siap. Saat terkoneksi dan berkomunikasi dengan *device* DTR perlu beri logika 1.
- e. *Data Set Ready (DSR)* berfungsi untuk menerima informasi status devais bahwa devais siap utnuik diakses oleh komputer melalui UART.
- f. *Request to Send (RTS)* berfungsi sebagai isyarat permintaan UART ke devais untukmemfasilitasi bahwa UART akan mengirimkan data ke devais.
- g. *Clear to Send (CTS)* berfungsi sebagai penerima jawaban atas pengiriman isyarat RTS bila modem/piranti telah menerima data.
- h. *Data Carrier Detect (DCD)* berfungsi sebagai penerima isyaratagar komputer bersedia menerima data pada pada waktu tertentu.

- i. *Ring Indicator (RI)* berfungsi menerima isyarat dari modem bahwa ada devais (eksternal) yang membutuhkan koneksi dalam rangka pengiriman atau permintaan data.

2.14 *Fan DC 12V*

Exhaust fan merupakan salah satu jenis kipas angin yg difungsikan untuk sirkulasi udara dalam ruang atau rumah. *Exhaust fan* berfungsi untuk menghisap udara di dalam ruang untuk dibuang ke luar, dan pada saat bersamaan menarik udara segar di luar ke dalam ruangan.



Gambar 2.10 *Exhaust Fan DC*

Selain itu *exhaust fan* juga bisa mengatur volume udara yang akan disirkulasikan pada ruang [26].

Spesifikasi *exhaust fan* yang perlu diperhatikan diantaranya:

- a. Konsumsi listrik (watt). Sesuaikan dengan daya sambungan listrik PLN kita. Kalau terlalu besar, kadang malah bikin jeblok.
- b. RPM yaitu rotation per minute atau putaran kipas per menit. Semakin tinggi RPM, semakin cepat sebuah *exhaust fan* menarik udara.
- c. *Noise* atau tingkat keberisikan suara *exhaust fan* dalam satuan desibel (db).
- d. *Air volume*, yaitu volume udara yang mampu ditarik oleh *exhaust fan*. Volume udara biasanya ditulis dalam satuan CMM (meter kubik per menit) atau CMH (meter kubik per jam).

2.15 *Power supply 12 VDC 2A*

Power supply adalah bagian pencatu daya bagi rangkaian elektronika. Fungsi utama rangkaian *power supply* adalah

mengubah tegangan AC jala-jala listrik menjadi tegangan DC yang dibutuhkan. alah satu jenis *Power supply* yang lazim digunakan yakni jenis *power supply* SMPS. *Switch Mode Power supply* (SMPS) karena sistem kerjanya menggunakan metode switching (pensaklaran) yaitu menghidup matikan tegangan yang masuk ke dalam trafo dengan peralatan/komponen elektronik dengan frekuensi tertentu. Sedangkan nama AC-matic diambil dari salah satu kelebihan dari SMPS yaitu kemampuan *power supply* bekerja dengan rentang tegangan masukan yang lebar. Pada beberapa jenis smps, mampu bekerja pada tegangan masukan antara 90 s/d 265V dengan output yang sama dan stabil. Karena kelebihan tersebut, smps menjadi *auto-voltage regulator* atau *wide range input regulated power supply* (secara Mudahnya Disebut AC-matic) [27].



Gambar 2.11 *Power supply Switching 12VDC*

Cara Pemakaiannya:

- a. Kabel Listrik dihubungkan ke lambang (N&L) Input AC
- b. Lambang -V dihubungkan 12V- (min)
- c. Lambang +V dihubungkan 12V+ (plus)
- d. Lambang V ADJ untuk mengatur tegangan DC supaya tepat menjadi 12 VDC

2.16 MCB (*Miniature Circuit Breaker*)

MCB (Miniature Circuit Breaker) atau Miniatur Pemutus Sirkuit adalah sebuah perangkat elektromekanikal yang berfungsi sebagai pelindung rangkaian listrik dari arus yang berlebihan. Dengan kata lain, MCB dapat memutuskan arus listrik secara otomatis ketika arus listrik yang melewati MCB tesebut melebihi nilai yang ditentukan. Namun saat arus dalam kondisi normal, MCB dapat berfungsi sebagai saklar yang bisa menghubungkan atau memutuskan arus listrik secara manual.

MCB pada dasarnya memiliki fungsi yang hampir sama dengan Sekering (FUSE) yaitu memutuskan aliran arus listrik rangkaian ketika terjadi gangguan kelebihan arus. Terjadinya kelebihan arus listrik ini dapat dikarenakan adanya hubung singkat (*Short Circuit*) ataupun adanya beban lebih (*Overload*). Namun MCB dapat diaktifkan kembali ketika rangkaian listrik sudah normal, sedangkan *Fuse/Sekering* yang terputus akibat gangguan kelebihan arus tersebut tidak dapat digunakan lagi [28].



Gambar 2.12 MCB Shukaku 220 V 2A

Biasanya MCB digunakan oleh pihak PLN untuk membatasi arus sekaligus sebagai pengaman dalam suatu instalasi listrik. MCB berfungsi sebagai pengaman hubung singkat (konsleting) dan juga berfungsi sebagai pengaman beban lebih. MCB akan secara otomatis dengan segera memutuskan arus apabila arus yang melewatinya melebihi dari arus nominal yang telah ditentukan pada MCB tersebut. Arus nominal yang terdapat pada MCB adalah 1A, 2A, 4A, 6A, 10A, 16A, 20A, 25A, 32A dan lain sebagainya. Nominal MCB ditentukan dari besarnya arus yang bisa ia hantarkan, satuan dari arus adalah *Ampere*, untuk kedepannya hanya akan saya tulis dengan A. Jadi jika MCB dengan arus nominal 2 *Ampere* maka hanya perlu ditulis dengan MCB 2A.

2.17 IC 7805

IC LM 7805 (regulator) adalah untuk menstabilkan tegangan dari catu daya bila terjadi perubahan tegangan IC *regulator* ini berfungsi untuk menstabilkan tegangan 5 V dan dapat bekerja dengan baik jika tegangan input (V_{in}) lebih besar minimal 2,5 V

dari pada tegangan *output* (V_{out}). Biasanya perbedaan tegangan input dengan *output* yang direkomendasikan tertera pada datasheet komponen tersebut [29].



Gambar 2.13 IC 7805

Keterangan:

- a. Pin 1 = *Voltage in* (7V-35V)
- b. Pin 2 = *Ground*
- c. Pin 3 = *Voltage out* (5V)

2.18 ChibiOs

ChibiOs adalah real-time OS yang digunakan untuk *embedded system* dan mendukung banyak mikroprosesor termasuk stm32. Dengan *ChibiOs*, kita bisa membangun program-program untuk akuisisi data, control system ataupun pemrosesan sinyal. Selain *ChibiOs*, modul STM32 ini ‘mencuri’ dari perangkat lunak *open source* lainnya seperti stm32flash, compiler gcc-arm-mone-eabi dan masih banyak lagi [30].

ChibiOS/RT merupakan RTOS yang menggunakan bahasa pemrograman C dan C++.

Berikut merupakan fitur-fitur pada ChibiOS/RT antara lain:

- a. Perangkat lunak gratis dengan lisensi GPL3.
- b. Dirancang untuk aplikasi RTOS.
- c. Portable.
- d. *Preemptive scheduling*.
- e. Mempunyai 256 tingkat prioritas, di mana bisa terdapat dua atau lebih *task* dengan prioritas yang sama.
- f. *Round robin scheduling* untuk *task* dengan prioritas yang sama.

- g. Terdapat *task/thread*, *virtual timers*, *semaphores*, *mutexes*, *condvars*, *event flags*, *messages*, *mailboxes*, *I/O queues*.
- h. Perancangan bisa dilakukan pada PC dengan *Windows* atau *Linux*.
- i. Terdapat fungsi opsional *heap allocator subsystem* dan *memory pools allocator subsystem*.
- j. *Blocking* dan *non-blocking* jalur I/O dengan kemampuan *timeout* dan pembuat *event*.
- k. Hampir semua tertulis dalam bahasa C dengan sedikit bahasa *assembler* untuk *porting*.
- l. Terdapat *hardware abstraction layer* (HAL) yang mendukung untuk berbagai macam peralatan seperti, serial, ADC, CAN I2C, MAC, MMC, PWM, SPI, UART, uIP, lwIP, dan FatFs.

Sedangkan untuk menggunakan ChibiOS/RT diperlukan mikrokontroler yang memenuhi spesifikasi minimum sebagai berikut:

- a. Arsitektur minimum CPU dengan 8-bits.
- b. Mendukung untuk bahasa standar C89 dan C99.
- c. Mendukung untuk maskable interrupt sources.
- d. Memiliki RAM minimal sebesar 2 KB.
- e. Memiliki memori untuk program sebesar 16 KB [31].



Gambar 2.14 Logo ChibiOs

2.19 Qt Creator

Qt Creator merupakan *cross-platform C++ integrated development environment* yang merupakan bagian dari Qt SDK. Qt Creator mempunyai *debugger* dalam bentuk *visual* dan *layout*

GUI serta tempat perancangan form. Teks editornya editornya mempunyai mempunyai fasilitas *syntax highlighting* dan *autocompletion*. Sebuah *project* dapat meliputi: file-file yang digroup secara bersama-sama, langkah-langkah *build* program, *form-form* dan file-file *resource*, dan pengaturan untuk menjalankan aplikasi. Proyek dapat dibuat secara manual atau diimport dari file proyek yang sudah ada. *Qt Creator* diinterraskan dengan sistem *cross-platform* untuk mem-build secara otomatis: *qmake* dan *CMake*. Proyek yang tidak menggunakan *qmake* atau *CMake* dapat diimport-kan, dan *Qt Creator* dapat meng-*ignore* sistem *build*.

Karena *Qt Creator* adalah sebuah *Integrated Development Enviroment Enviroment (IDE)*, maka *Qt Creator* memisahkan memisahkan antara text editor untuk *build* dan editor untuk menjalankan (run) aplikasi-aplikasi. *Qt Creator* bukan hanya bisa membaca *text file* biasa, akan tetapi juga bisa membaca file C++ dan bahasa *QML* [32].



Gambar 2.15 Tampilan *Qt-Creator*

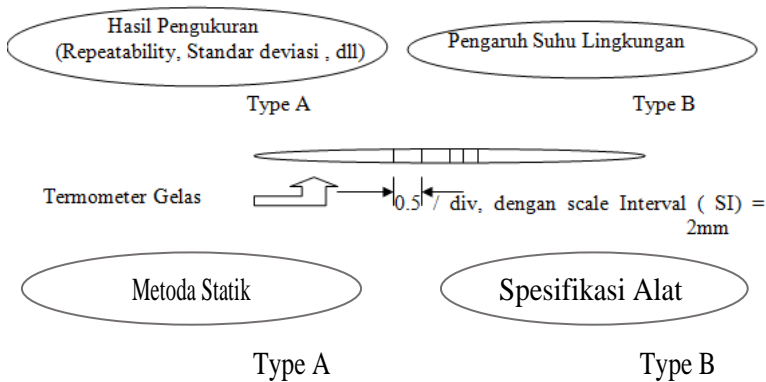
2.20 STM32 ST-LINK

STM32 ST-LINK Utility (STSW-LINK004) adalah antarmuka perangkat lunak berfitur lengkap untuk pemrograman mikrokontroler STM32. Ini menyediakan lingkungan yang mudah digunakan dan efisien untuk membaca, menulis dan memverifikasi perangkat memori. Alat ini menawarkan berbagai fitur untuk memprogram memori internal STM32 (*Flash*, *RAM*,

OTP dan lainnya), kenangan eksternal, untuk memverifikasi konten pemrograman (*checksum*, verifikasi selama dan setelah pemrograman, bandingkan dengan file) dan untuk mengotomatisasi pemrograman STM32 [33].

2.21 Analisa Sumber-Sumber Ketidakpastian

Sumber-sumber ketidakpastian yang turut memberikan kontribusi selain ada pada diri manusia sendiri sebagai pelaku pengukuran / kalibrasi juga pada alat-alat bantu (kalibrator) yang digunakan untuk mengukur suhu pasien tersebut, juga resolusi alatnya, pengaruh suhu lingkungan. Secara rinci dari sumber-sumber ketidakpastian dapat digambarkan sebagai berikut:



Gambar 2.16 Diagram Alir Penentuan Nilai Ketidakpastian Baku dari Data Tipe A dan B

Untuk mengevaluasi masing-masing sumber ketidakpastian tersebut, diperlukan analisa dengan menggunakan metoda statistik, yang disebut analisa *type A*, dan analisa *type B* berdasarkan adanya sertifikat kalibrasi atau tidak dan spesifikasi dari alat tersebut. untuk lebih jelasnya dapat dilihat sebagai berikut:

a. Analisa *Type A*, (U_a)

Pada tipe ini biasanya ditandai dengan adanya data pengukuran, misalnya n kali pengukuran, maka selanjutnya dari data tersebut, akan ditemukan nilai rata-ratanya, standar deviasinya, dan atau *repeatability*-nya. Bentuk kurva dari tipe ini adalah sebaran *Gauss*. Rumus umum ketidakpatian untuk tipe A ini adalah:

$$U_{A1} = \frac{\sigma}{\sqrt{n}} \dots\dots\dots (2.2)$$

Dimana:

σ = Standar deviasi

n = Banyaknya data

Rumus standar deviasi (σ) sendiri adalah sebagai berikut:

$$\sigma = \frac{\sqrt{\sum (y_i - \bar{y})^2}}{n-1} \dots\dots\dots (2.3)$$

Dimana:

y_i = nilai koreksi ke- i

\bar{y} = rata-rata nilai koreksi

σ = Standard Deviasi

Sedangkan untuk U_{A2} rumusnya dapat diketahui seperti di bawah ini:

$$U_{A2} = \sqrt{\frac{SSR}{n}} \dots\dots\dots (2.4)$$

Dimana:

SSR (*Sum Square Residual*) = $\sum SR$ (*Square Residual*)

SR = R^2 (*Residu*)

$$SSR = \sum R^2 \dots\dots\dots (2.5)$$

$$R = Y_i - Y_{\text{Reg}} \dots\dots\dots (2.6)$$

$$Y_{\text{Reg}} = a + bX_i \dots\dots\dots (2.7)$$

$$a = \bar{y} + (b\bar{x}) \dots\dots\dots (2.8)$$

$$b = \frac{n \cdot \sum x_i y_i - \sum x_i \sum y_i}{n \cdot \sum x_i^2 - (\sum x_i)^2} \dots\dots\dots (2.9)$$

b. Analisa *Type B* (U_b)

Pada analisa tipe ini akan digunakan selain metode statistik, yaitu berdasarkan adanya sertifikat kalibrasi atau tidak dan spesifikasi dari alat tersebut. Berhubung dalam laporan ini alat ukur standar yang dipakai tidak ada sertifikat kalibrasi, maka rumusnya adalah sebagai berikut:

$$U_{B1} = \frac{a}{\sqrt{3}} \dots\dots\dots (2.10)$$

$$U_{B2} = \frac{0,5 \text{ Resolusi}}{\sqrt{3}} \dots\dots\dots (2.11)$$

Dimana:

U_{B1} = Ketidakpastian dari alat standar

U_{B2} = Ketidakpastian resolusi

c. Ketidakpastian Kombinasi (U_c)

Selanjutnya dari semua sumber ketidakpastian tersebut harus dikombinasikan atau digabungkan untuk memberikan gambaran menyeluruh ketidakpastian dari hasil kalibrasi tersebut. Rumus umum ketidakpastian kombinasi adalah:

$$U_c = \sqrt{\sum (U_a)^2 + \sum (U_b)^2} \dots\dots\dots (2.12)$$

Atau secara umum:

$$U_c^2 = \sum (C_i \cdot U_i)^2 \dots\dots\dots (2.13)$$

Dengan C_i = Koefisien sensitifitas dari ketidakpastian ke-
i

d. Derajat Kebebasan Efektif (V_{eff})

Nilai faktor cakupan, k untuk perkalian ketidakpastian diperluas diatas didapat dari derajat kebebasan efektif, V_{eff} , dengan rumus:

$$V_{eff} = \frac{(U_c)^4}{\sum (U_i)^4 / v_i} \dots\dots\dots (2.14)$$

Dengan:

U_c = Ketidakpastian kombinasi/gabungan

U_i = Ketidakpastian individual ke-i

V_i = Derajat kebebasan pada ketidakpastian individual ke-i

e. Tingkat Kepercayaan (U_{95})

Tingkat kepercayaan merupakan tingkatan keyakinan akan keberadaan nilai sebenarnya pada suatu tindak pengukuran dengan menggunakan alat tertentu.

f. Faktor Cakupan (k)

Faktor cakupan merupakan faktor pengali pada ketidakpastian, sehingga membentuk cakupan logis pada penggunaan keseharian. Faktor cakupan dicari menggunakan tabel *T-Student Distribution*

g. Ketidakpastian Diperluas (U_{exp})

Dalam pelaporan ketidakpastian hasil pengukuran/kalibrasi yang dilaporkan adalah ketidakpastian yang

sudah dalam perluasan (*expanded*), sehingga hasil tersebut sangat logis dalam kenyataan, selain itu dengan menggunakan tingkat kepercayaan 95 %, seperti lazimnya dipakai dalam pelaporan-pelaporan saat ini, lain halnya jika ada pengecualian dengan mengambil tingkat kepercayaan tertentu [34]. Rumus ketidakpastian diperluas (*expanded uncertainty*) adalah:

$$U_{\text{exp}} = k U_c \dots\dots\dots (2.15)$$

Dengan:

U_{95} = Ketidakpastian diperluas

k = Faktor Cakupan

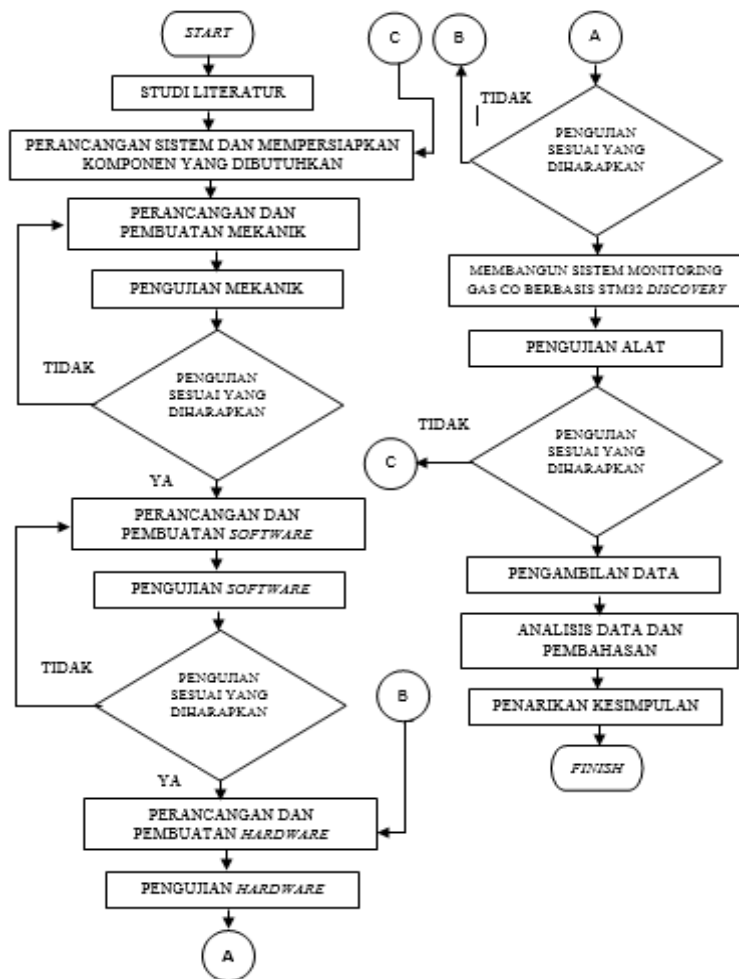
U_c = Ketidakpastian kombinasi

Halaman Ini Sengaja Dikosongkan

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 *Flowchart* dan Diagram Blok Perancangan Alat

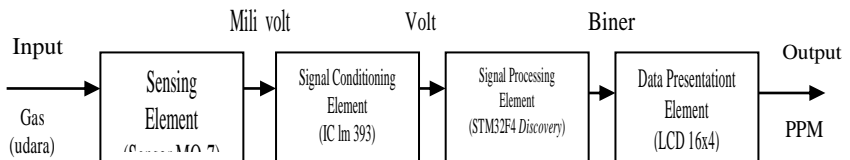
Berikut tahapan tugas akhir digambarkan dalam flowchart :



Gambar 3.1 *Flowchart* Pengerjaan Tugas Akhir

Flowchart diatas merupakan flowchart tentang tahap pengerjaan tugas akhir dari *start* hingga *finish*. Tahap awal pada flowchart pengerjaan tugas akhir ini dimulai dengan adanya studi literatur untuk mencari referensi atau materi yang menunjang tugas akhir mengenai “Rancang Bangun Sistem Monitoring Konsentrasi Gas Karbon monoksida (CO) Sebagai Emisi Gas Buang Menggunakan Sensor Gas MQ-7 Berbasis Mikrokontroler STM32F4 *Discovery*”. Setelah melakukan studi literatur, selanjutnya adalah melakukan perancangan sistem dan mempersiapkan komponen yang dibutuhkan dalam perancangan sistem. Kemudian membuat perancangan mekanik, *software* dan *hardware* dari sistem monitoring konsentrasi gas karbon monoksida (CO) berbasis STM32F4 *Discovery*. Apabila dalam perancangan tersebut sudah sesuai yang diharapkan, maka rancang bangun sistem *monitoring* gas siap dilakukan untuk pengujian alat. Pengujian alat dilakukan dengan memberikan input berupa beberapa sampel uji gas yang mengandung kandungan karbonmonoksida (CO) untuk mengetahui apakah sensor MQ 7 sudah dapat bekerja dengan baik. Apabila semua rancang bangun sistem monitoring gas dapat bekerja sesuai dengan yang diharapkan, maka selanjutnya dilakukan pengambilan data pada rancang bangun *monitoring* gas. Kemudian melakukan analisis data menggunakan hasil uji performansi dan kalibrasi alat. Setelah itu dapat dilakukan penarikan kesimpulan yang merupakan proses terakhir dalam flowchart pengerjaan alat ini.

Pada perancangan sistem *monitoring* konsentrasi gas karbonmonoksida (CO) ini terdapat diagram blok pengukuran. Adapun diagram blok sistem pengukuran pada gambar 3.2



Gambar 3.2 Diagram Blok Sistem Monitoring Gas

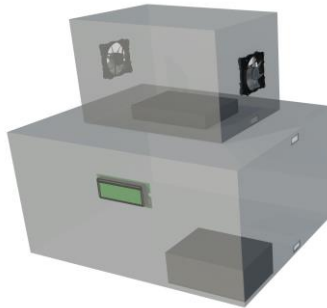
Gambar 3.2 di atas merupakan gambar diagram blok sistem pengukuran yang dilakukan oleh sensor. Mulai dari input, sensor (*sensing element*), pengkondisian sinyal (*signal conditioning element*), pemrosesan sinyal (*processing signal element*), dan penampil data (*data presentation*).

Pada sistem monitoring konsentrasi gas ini menggunakan sensor MQ-7. MQ-7 berfungsi sebagai sensor untuk mengukur konsentrasi gas karbonmonoksida (CO). Pada diagram blok pengukuran diatas, sensor tersebut termasuk *sensing element*. Sensor MQ-7 dirangkai pada sebuah modul sensor, yang nantinya *output* dari sensor MQ-7 sudah merupakan *digital output*. Setelah mendapatkan keluaran digital output, maka tahap selanjutnya adalah pemrosesan sinyal. Pemrosesan sinyal dilakukan oleh mikrokontroler STM32F4 *Discovery*. Sinyal *digital* yang diterima adalah berupa bilangan digital dimana bilangan digital ini menunjukkan pengukuran konsentrasi gas karbonmonoksida (CO). Maka, agar dapat ditampilkan dalam angka, sinyal tersebut diolah di mikrokontroler STM32F4 *Discovery*. Setelah diolah untuk penampilan data digunakan LCD (*Liquid Crystal Display*).

3.2 Desain Rancang Bangun Sistem Monitoring Gas CO

Sistem monitoring emisi gas buang ini terdiri 4 sensor pengukuran konsentrasi gas karbon monoksida (CO), sulfur dioksida (SO₂), karbon dioksida (CO₂), dan Nitrogen dioksida (NO₂). Desain alat terdiri dari 2 kotak, yakni kotak atas dan kotak bawah. Kotak atas memiliki ukuran 15 x 15 x 15 cm dan menggunakan bahan dari *acrylic*. Pada kotak atas terdapat rangkaian sensor dan *fan dc*. Dimana kotak atas berfungsi sebagai tempat pengambilan udara yang telah dikondisikan menggunakan dua arah *fan dc* berbeda (mengambil dan mengeluarkan udara). Kotak bawah memiliki ukuran 30 x 30 x 30 cm dan menggunakan bahan dari *acrylic*. Kotak bawah terdapat komponen-komponen yang digunakan dalam rancang bangun sistem monitoring gas ini, seperti STM32F4 *Discovery*, LCD 16x4, Modul RTC, Modul SD Card, Modem Wavecom M1306B

serial RS232, dan rangkaian *power supply*. Berikut ini desain awal perancangan alat pada gambar 3.3



Gambar 3.3 Desain Alat pada *Software Sketch Up*

3.3 Perancangan Modul Sensor Gas MQ-7

Pada sistem monitoring gas karbon monoksida (CO) menggunakan modul sensor MQ-7. Dimana modul sensor MQ-7 ini memiliki dua fungsi yakni sebagai *sensing element* dan pengkondisian sinyal yang sudah terpasang didalam rangkaian modul sensor. *Sensing element* dilakukan oleh sensor MQ-7 dan pengkondisian sinyal dilakukan oleh IC LM 393.

Sensor Gas MQ-7 merupakan *sensor* yang digunakan untuk mengetahui kadar atau konsentrasi gas karbonmonoksida (CO), sensor ini bisa mengukur dengan *range* 10~500ppm CO, dan tegangan *output* berada di *range* 2.5V~4.3V (*in* 150ppm CO). Standar kondisi kerja sensor dengan temperatur sebesar $-20^{\circ}\text{C} \pm 2^{\circ}\text{C}$.

Sensor ini yang bereaksi terhadap kadar gas karbon monoksida yang terdapat dalam udara. Modul ini memiliki keluaran data analog serta desain *hardware* minimalis yang ditujukan untuk memudahkan proses penggunaan sensor MQ-7. Modul ini dapat diaplikasikan sebagai *alarm* peringatan dini, ataupun gas *detector* untuk membantu proses industri yang melibatkan gas karbon monoksida. Berikut ini skematik sensor gas MQ-7 pada gambar 3.4



Gambar 3.4 Skematik CO Gas Sensor Module

Pada modul sensor MQ-7 ini terdapat 4 pin keluaran. Pada pin 1 merupakan Vin yang berfungsi sebagai tegangan sumber untuk menhidupkan sensor. Pada pin 2 merupakan Aout yang berfungsi sebagai keluaran *analog* dari sensor. Pada pin 3 merupakan Dout yang berfungsi sebagai keluaran *digital* dari sensor. Pada pin 4 merupakan *Ground*. Dalam perancangan yang telah dibuat, modul sensor MQ-7 dihubungkan kedalam STM32F4 *Discovery* yang telah terhubung dengan *shield* yang sudah dibuat. Pin Vin pada modul dihubungkan ke VCC pada *shield* STM32F4 *Discovery*. Pin *Ground* dihubungkan ke *ground* pada *shield* STM32F4 *Discovery*. Pin Aout dihubungkan ke pin PA0 pada *shield* STM32F4 *Discovery*. Berikut ini peletakkan sensor MQ-7 pada gambar 3.5

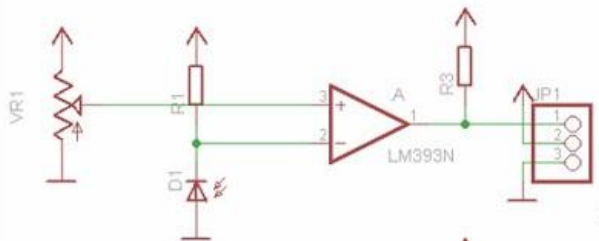


Gambar 3.5 Modul sensor MQ-7

Disisi lain, pengkondisian sinyal dalam modul sensor MQ-7 menggunakan IC LM 393. IC LM 393 adalah IC pembanding tegangan (*voltage comparator*) yang dapat beroperasi dengan

catu daya tunggal maupun daya *double*. IC ini dirancang untuk memungkinkan rentang tegangan dari moda *common-to-ground* pada operasi IC LM 393. Dalam satu kemasannya mempunyai dua buah komparator didalamnya. IC ini berfungsi untuk membandingkan dua macam tegangan yang terdapat pada kedua inputnya. Dalam aplikasinya biasanya salah satu pin input dari komparator sebagai tegangan referensi sedangkan pin input lainnya sebagai tegangan yang akan dibandingkan.

Dalam penggunaan di modul MQ &, IC LM 393 berfungsi untuk membandingkan tegangan yang dipakai. Keluaran dari sensor MQ 7 berupa 250 milivolt sehingga untuk masuk kedalam mikrokontroler tegangan tersebut harus dinaikkan sebesar 3,3-volt. Berikut ini skematik IC LM 393 MQ-7 pada gambar 3.6



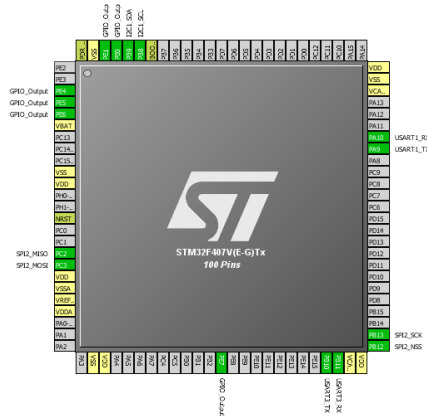
Gambar 3.6 Skematik IC LM 393

Pada rangkaian tersebut tegangan referensinya diperoleh dari sebuah VR (*Variable Resistor*) dan tegangan yang akan dibandingkan berasal dari sensor MQ-7 yang dirangkai menjadi rangkaian pembagi tegangan. Dengan tegangan referensi dari VR maka *output* dari komparator dapat diatur "pada konsentrasi berapa *output* dari regulator akan bernilai nol". Dalam aplikasinya *output* dari komparator LM 393, membutuhkan resistor *pullup* dengan tegangan V+ yaitu untuk menjaga tegangan *output* supaya memiliki logika satu ketika kondisi diam.

3.4 Perancangan *Shield Board STM32F4 Discovery*

Mikrokontroler yang digunakan pada perancangan sistem monitoring gas ini menggunakan *STM32F4 Discovery*. Dimana

mikrokokontroller ini memiliki peran yang penting dalam menjalankan sistem dan menjadi kontroller utama dalam kerja sistem. Dalam penggunaan STM32F4 *Discovery* ini telah diatur pin-pin yang digunakan sesuai fungsinya. Berikut konfigurasi fungsi pin yang digunakan dalam perancangan sistem pada gambar 3.7



Gambar 3.7 Konfigurasi pin STM32F4 *Discovery*

Tabel 3.1 Konfigurasi pin STM32F4 *Discovery*

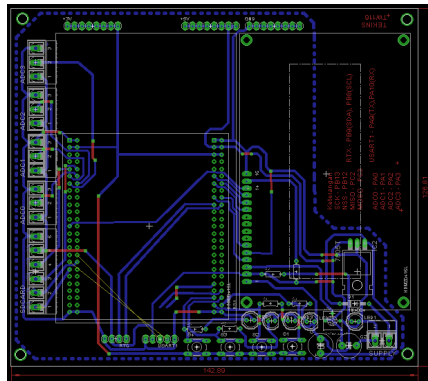
Pin	Fungsi Alternatif	Keterangan
PA0	USART2_CTS/USART4_TX/ETH_MII_CRSDV/ETH_MII_TXD3/TIM2_CH1_ETR/TIM5_CH1/TIM8_ETR/ADC123_IN0/WKUP	ADC CO
PA1	USART2_RTS/USART4_RX/ETH_RMII_RETX_F_CLK/ETH_MII_RX_CLK/TIM5_CH2/TIM2_CH2/ADC123_IN1	ADC SO ₂
PA2	USART2_TX/TIM5_CH3/TIM9_CH1/TIM2_CH3/ETH_MDIO/ADC123_IN2	ADC CO ₂
PA3	USART2_RX/TIM5_CH4/TIM9_CH2/TIM2_CH4/OTG_HS_ULPI_D0/ETH_MII_COL/ADC123_IN3	ADC NO ₂
PB8	TIM4_CH3/SDIO_D4/TIM10_CH1/DCMI_D6/OTG_FS_SCL/ETH_MII_TXD3/I2C1_SCL	I2C SCL

	CAN1_RX	
--	---------	--

Tabel 3.1 (Lanjutan)

PB9	SPI2_NSS/I2S2_WS/TIM4_CH4/TIM11_CH1/ OTG_FS_SDA/SDIO_D5/DCMI_D7/ I2C1_SDA/CAN1_TX	I2C SDA
PB10	SPI2_SCK/I2S2_CK/I2C2_SCL/USART3_TX/ OTG_HS_ULPI_D3/ETH_MII_RX_ER/OTG _HS_SCL/TIM2_CH3	USART TX
PB11	I2C2_SDA/USART3_RX/OTG_HS_ULPI_D4/ ETH_RMII_TX_EN/ETH_MII_TX_EN/ OTG_HS_SDA/TIM2_CH4	USART RX
PB12	SPI2_NSS/I2S2_WS/I2C2_SMBA/ USART3_CK/TIM1_BKIN/CAN2_RX/OTG_ HS_ULPI_D5/ETH_RMII_TXD0/ETH_MII_ TXD0/OTG_HS_ID	SPI NSS
PB13	SPI2_SCK/I2S2_CK/USART3_CTS/ TIM1_CH1N/ CAN2_TX/OTG_HS_ULPI_D6/ ETH_RMII_TXD1/ETH_MII_TXD1/ OTG_HS_VBUS	SPI SCK
PC2	SPI2_MISO/OTG_HS_ULPI_DIR/ TH_MII_TXD2/I2S2ext_SD/ ADC123_IN12	SPI MISO
PC3	SPI2_MOSI/I2S2_SD/ OTG_HS_ULPI_NXT/ETH_MII_TX_CLK/ ADC123_IN13	SPI MOSI
PE0	TIM4_ETR/ FSMC_NBL0/ DCMI_D2	OUTPUT LCD R/S
PE1	FSMC_NBL1/ DCMI_D3	OUTPUT LCD E/N
PE4	TRACED1/ FSMC_A20/ DCMI_D4	OUTPUT LCD D4
PE5	TRACED2/ FSMC_A21/ TIM9_CH1/ DCMI_D6	OUTPUT D5
PE6	TRACED3/ FSMC_A22/ TIM9_CH2/ DCMI_D7	OUTPUT D6
PE7	FSMC_D4/ TIM1_ETR	OUTPUT D7

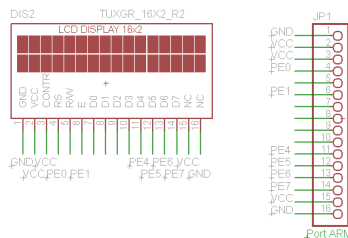
Shield board dirancang untuk memudahkan dalam penyambungan komponen satu dengan yang lainnya dalam pin-pin STM32F4 *Discovery* dengan beberapa komponen seperti 4 sensor gas meliputi sensor gas MQ 7, MQ 136, MG 811, MQ 135, LCD 16x4, RTC DS3231, *Modul SD Card*, *Modul Wavecom*. Berikut skematik *shield* yang digunakan dalam perancangan pada gambar 3.8



Gambar 3.8 Skematik *Shield Board* STM32F4 *Discovery*

3.5 Perancangan *Liquid Crystal Display* (LCD)

Rangkaian skematik *Liquid Crystal Display* (LCD) ke STM32F4 *Discovery* pada gambar 3.9



Gambar 3.9 Rangkaian Skematik LCD pada STM32F4 *Discovery*

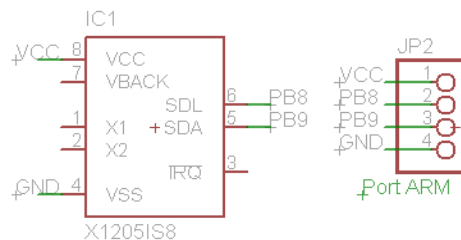
LCD merupakan suatu *modul* penampil. Dalam hal ini untuk menampilkan data yang terdeteksi pada sensor sehingga dapat ditampilkan data berupa *digital* yang menunjukkan *display* nilai konsentrasi gas CO berupa nilai ADC dan PPM pada hasil pengukuran sensor. LCD yang digunakan dalam rancang bangun sistem *monitoring* gas ini adalah LCD 16 kolom x 4 baris. Berikut tampilan LCD pada sistem monitoring pada gambar 3.10



Gambar 3.10 Penempatan LCD untuk *Display*

3.6 Perancangan *Real Time Clock (RTC)*

Pada perancangan sistem monitoring gas ini, terdapat RTC eksternal yakni Modul RTC DS3231 dengan baterai litium. RTC ini berfungsi sebagai input pemberi referensi waktu terhadap data yang akan diperoleh. Cara kerjanya adalah alamat dan data ditransmisikan secara serial melalui sebuah jalur data dua arah I2C. Akses modul ini dilakukan melalui antarmuka I2C yang dapat dikatakan identik dengan pengalamatan *register* pada RTC DS1307, sehingga kode *program* yang sudah dibuat untuk mikrokontroler lain dapat berjalan tanpa perlu dimodifikasi. Berikut perancangan skematik RTC pada gambar 3.11



Gambar 3.11 Skematik Real Time Clock (RTC) pada STM32F4 *Discovery*

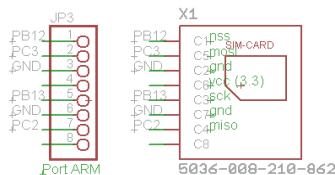
RTC ini hanya memerlukan dua buah pin saja untuk berkomunikasi yaitu pin data dan pin untuk sinyal *clock* (SDA dan SCL) sehingga STM32F4 *Discovery* dapat mengolah data dan clock yang diterima dari RTC untuk dijadikan referensi waktu. Pin SDA dihubungkan ke port PB9 dan pin SCL ke pin PB8 STM32F4 *Discovery*. Berikut ini pemasangan RTC pada shield board STM32F4 *Discovery* pada gambar 3.12



Gambar 3.12 Modul RTC DS3231

3.7 Perancangan Modul SD Card

Perancangan modul *sd card* digunakan dalam sistem monitoring ini. Modul *SD card* digunakan sebagai *data logger* dalam penyimpanan informasi dari hasil ppm yang telah diukur. Dalam penggunaan modul *sd card* ini, menggunakan *sd card* dengan memori sebesar 8 gb. Penyimpanan data ini disimpan dalam format *excel* atau *.csv* sehingga dapat mudah dilihat. Terdapat 6 pin dalam *modul sd card* ini yakni *Ground*, *VCC*, *MISO*, *MOSI*, *SCK*, dan *CS*. Berikut perancangan skematik modul *sd card* pada gambar 3.13



Gambar 3.13 Skematik Modul SD Card pada STM32F4 *Discovery*

Dalam perancangannya, menghubungkan pin pada *modul SD Card* kedalam mikrokontroller STM32F4 *Discovery* yang terhubung dengan shield yang telah dibuat. Pin *Ground* pada modul disambungkan ke *Ground* STM32F4 *Discovery*. Pin *VCC* pada *modul SD Card* disambungkan ke pin 5 volt STM32F4 *Discovery*. Pin *MISO* pada *modul SD Card* disambungkan ke pin PC2 STM32F4 *Discovery*. Pin *MOSI* pada *modul SD Card* disambungkan ke pin PC3 STM32F4 *Discovery* . Pin *SCK* pada *modul SD Card* disambungkan ke pin PB13 STM32F4 *Discovery* dan Pin *CS* pada *modul SD Card* disambungkan ke pin PB12 STM32F4 *Discovery*. Berikut pemasangan *modul sd card* pada *shield board* STM32F4 *Discovery* pada gambar 3.14

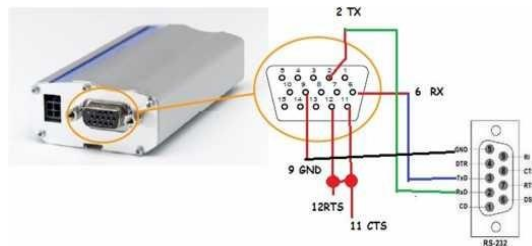


Gambar 3.14 Modul *SD Card*

3.8 Perancangan *Modem Wavecom*

Pada bagian rancang bangun monitoring gas ini terdapat modem wavecom M1306B serial RS232. Dimana modem ini banyak digunakan oleh pengguna layanan sms gateway untuk broadcast sms, kirim sms massal dan *compatible* dengan engine sms seperti gammu dan quick gateway. Modem ini juga dilengkapi dengan *AT Command* untuk semua jenis operasi karena mendukung fasilitas koneksi RS232 dan dapat dengan cepat terhubung ke *port serial* komputer maupun mikrokontroller sehingga sangat mudah untuk dikendalikan. Kelas ini juga mendukung 10 tingkat kecepatan *transfer data*.

Dalam perancangan penggunaan *modem wavecom M1306B serial RS232* dilakukan dengan cara mengambil pin Rx, Tx, dan Ground dari *serial RS232 modem wavecom M1306B* dan menghubungkan pin Rx, Tx, dan Ground ke dalam mikrokontroler STM32F4 *Discovery* yang terhubung dengan *shield* yang telah dibuat. Berikut rangkaian modem wavecom *serial RS232* pada gambar 3.15



Gambar 3.15 Konfigurasi *Pin Modem Wavecom*

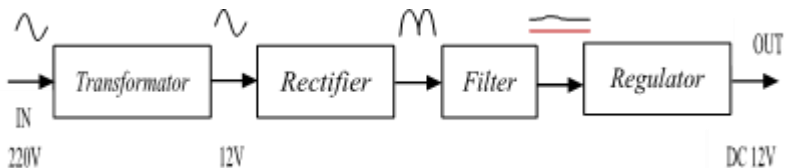
Pin *Ground* pada *modul SD Card* disambungkan ke *Ground* STM32F4 *Discovery*. Pin *VCC* pada *modul SD Card* disambungkan ke pin 5 volt STM32F4 *Discovery*. Pin Rx pada *modem wavecom* disambungkan ke pin PB11 STM32F4 *Discovery*. Pin Tx pada *modem wavecom* disambungkan ke pin PB10 STM32F4 *Discovery*. Karena tegangan keluaran dari modem wavecom tidak compatible dengan mikrokontroller maka harus dihubungkan dengan modul RS232 agar sesuai dengan tegangan pada mikrokontroller. Berikut pemasangan *modem wavecom* pada *shield board* STM32F4 *Discovery* pada gambar 3.16



Gambar 3.16 *Modem Wavecom M1306B Serial*

3.9 Rangkaian Power Supply

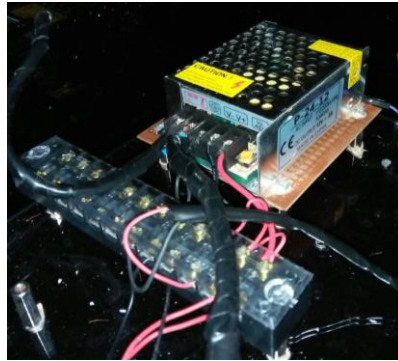
Power supply merupakan sumber tenaga yang dibutuhkan suatu rangkaian elektronika untuk bekerja. Besar *power supply* ini tergantung oleh spesifikasi dari alat masing-masing. Pada perancangan sistem monitoring ini *power supply* digunakan untuk menyediakan sumber tegangan untuk rangkaian mikrokontroler STM32F4 *Discovery*, rangkaian sensor MQ-7, modul RTC, Modul SD Card, modem wavecom serial RS232, dan LCD 16x4. Berikut blok diagram dari *power supply* 12VDC pada gambar 3.17



Gambar 3.17 Blok Diagram *Power Supply* 12VDC

Power supply yang digunakan dalam rancang bangun sistem monitoring tugas akhir ini menggunakan sumber langsung (input) dari PLN yakni 220VAC. Tegangan tersebut kemudian diturunkan menjadi 12V oleh *transformator*. Ketika tegangan sudah diturunkan, tegangan tersebut masih berupa tegangan AC, sehingga akan diubah menjadi tegangan DC oleh *rectifier*. Kemudian sinyal arus yang keluar dari *rectifier* diratakan (*smoothing*) oleh filter. Untuk menghasilkan tegangan dan arus DC

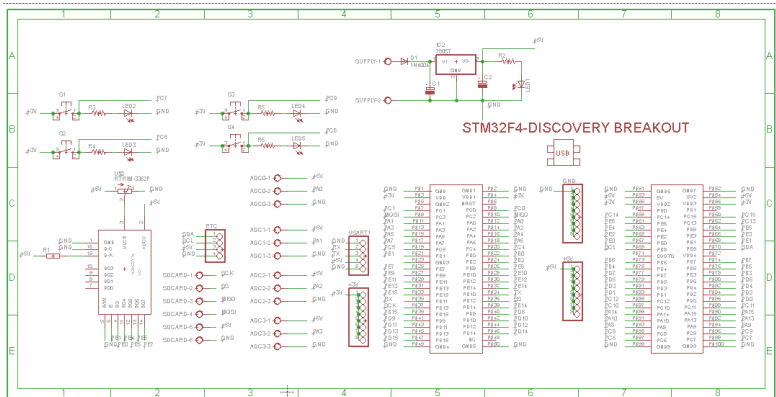
(searah) yang tetap dan stabil, maka diperlukan regulator. Regulator ini berfungsi untuk mengatur tegangan sehingga tegangan *output* tidak terpengaruh oleh suhu, arus, beban dan tegangan *input* yang berasal dari *output filter*. Berikut pemasangan *power supply 12VDC* pada gambar 3.18



Gambar 3.18 *Power Supply*

3.10 Perancangan dan Pembuatan *Software*

Pada perancangan dan pembuatan *software* dilakukan dengan menggunakan *software eagle*, yang berfungsi untuk mengintegrasikan *sensor MQ 7*, *SD Card*, *LCD 16x4*, *RTC*, *SMS Gateway* dan mikrokontroler *SM32F4 Discovery*. Berikut merupakan rangkaian skematik mikrokontroler *STM32F4*, sensor *MQ7*, *SD Card*, *RTC*, *UART (SMS Gateway)* dan *LCD 16x4* menggunakan *software eagle*. Berikut skematik dari perancangan sistem yang dibuat pada *software eagle* pada gambar 3.19

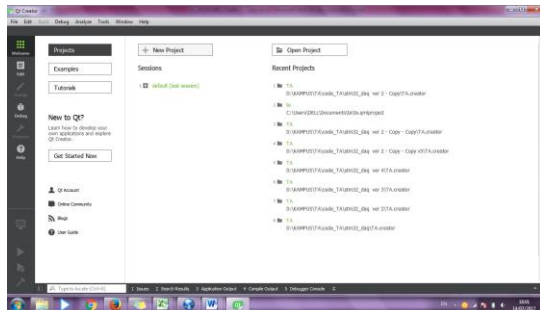


Gambar 3.19 Rangkaian Skematik Mikrokontroler STM32F4, sensor MQ-7, SD Card, RTC, Uart (SMS Gateway) dan LCD 16x4

Perancangan *software* pada tahap ini merupakan siklus gabungan untuk rangkaian pemrosesan sinyal. Rangkaian pemrosesan sinyal pada alat ini menggunakan mikrokontroler STM32F4 *Discovery*. Untuk membuat program pada STM32F4 *Discovery*, dibutuhkan *software* *ChibiOs*, *Qt Creator* untuk *editing* (menulis project), *notepad++* untuk mengcompile dan *ST-LINK* untuk mendownload program di mikrokontroler STM32F4 *Discovery*.

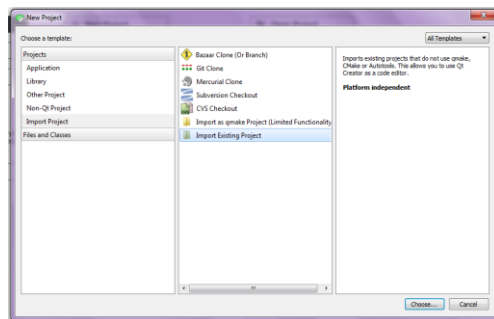
Kemudian untuk pemrograman menggunakan *software* *Qt-Creator* dilakukan dengan langkah-langkah sebagai berikut:

- Software* *Qt Creator* dibuka.
- Klik *New project*. Kemudian akan muncul gambar seperti dibawah ini.



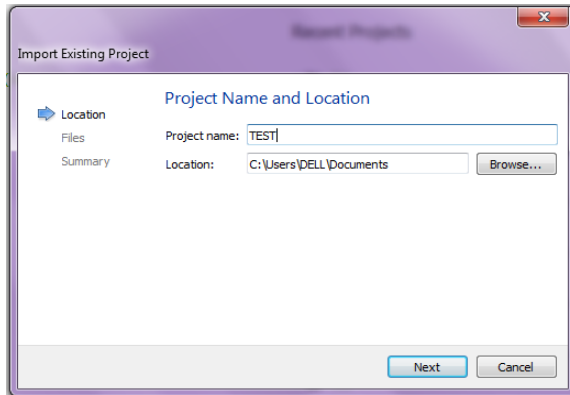
Gambar 3.20 *Create New Project*

- c. Pilih *import project*. Lalu klik *import existing project* seperti gambar dibawah ini:



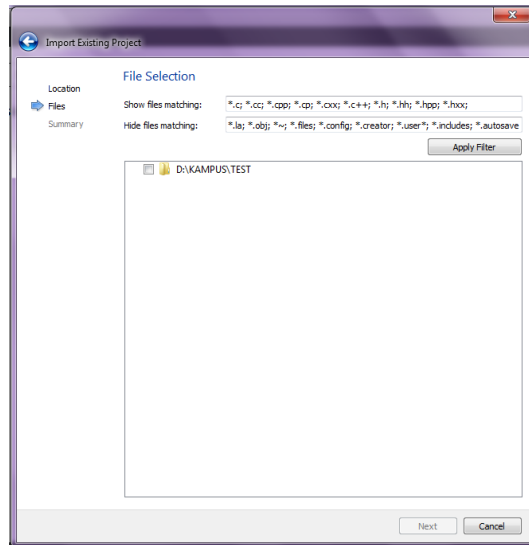
Gambar 3.21 *Project Qt Creator*

- d. Kemudian akan muncul dialog *import existing project*. Beri nama *project* sesuai yang diinginkan dan pilih lokasi yang diinginkan dalam laptop. Kemudian klik *next*.



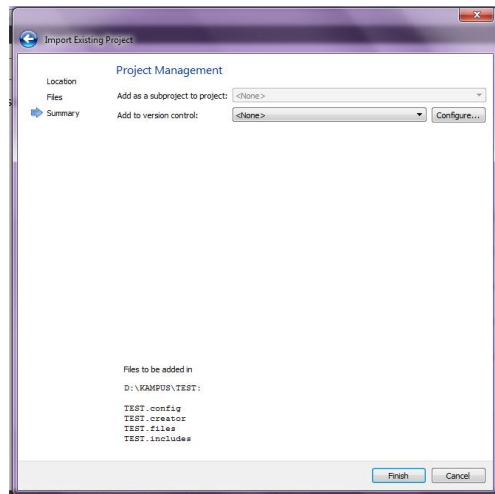
Gambar 3.22 *Import Existing Project Name and Location*

- e. Kemudian akan muncul kotak *file selection*. Setelah itu centang *location file* dan klik *next*.



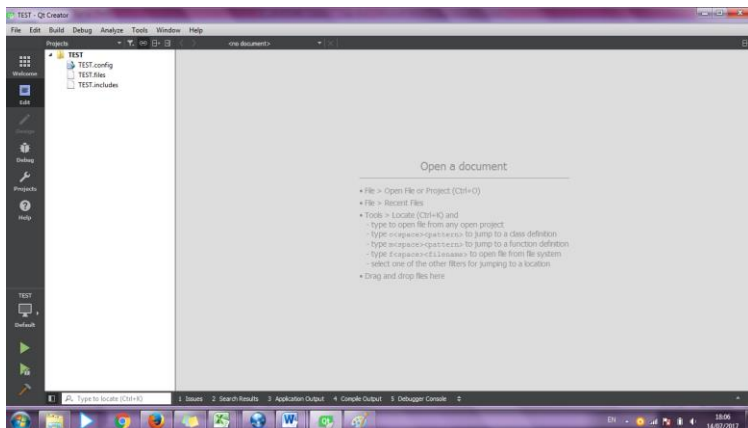
Gambar 3.23 *Import Existing Project File Selection*

- f. Setelah itu akan muncul kotak *project management*. Lalu klik *finish*.



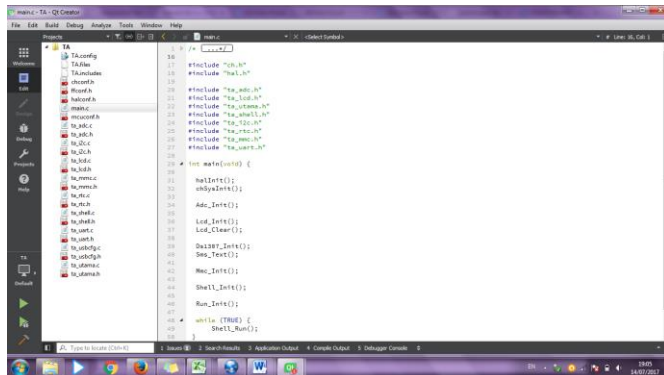
Gambar 3.24 *Import Existing Project Management*

- g. Setelah itu akan muncul *project* yang berisi *file* utama meliputi *.config*, *.files*, *.includes*.



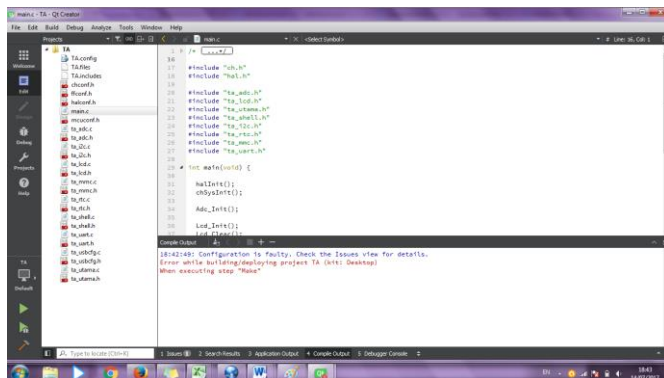
Gambar 3.25 *Tampilan Awal Program*

- h. Kemudian pada *project.files* diisi fungsi-fungsi yang digunakan sesuai gambar dibawah ini:



Gambar 3.28 *Class pada Project*

- k. Setelah itu akan muncul *program* yang telah diatur dan pemrograman bisa dilakukan.
- l. Setelah selesai menyusun *program*, program dapat di-*compile*.



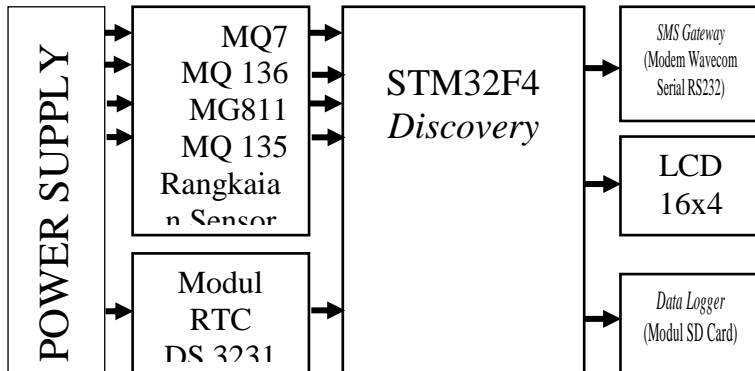
Gambar 3.29 *Build Project Qt Creator*

- m. *Software* Qt Creator dan PC tidak *compatible* sehingga diperlukan aplikasi tambahan berupa *notepad++* untuk meng-*compile*. Pada *notepad++* masukkan *project* yang dibuat (Makefile). Kemudian klik *tools* dan *make all*, seperti gambar dibawah ini:

BAB IV PENGUJIAN DAN ANALISIS DATA

4.1 Rancang Bangun Alat

Berikut ini alur sistem dari alat *Monitoring* gas yang telah dibuat pada gambar 4.1.



Gambar 4.1 Alur Sistem *Monitoring* Gas

Sistem *Monitoring* gas ini bekerja dengan mengaktifkan power supply terlebih dahulu. Ketika *power supply* telah aktif maka semua komponen yang sudah terintegrasi akan bekerja. *Fan DC* akan menarik udara dari luar untuk masuk kedalam kotak atas yang berisi empat sensor yakni sensor MQ7 (konsentrasi gas CO), MQ 136 (Konsentrasi gas SO₂), MG 811 (konsentrasi gas CO₂), dan MQ 135 (Konsentrasi gas NO₂). *Fan DC* yang lainnya berfungsi untuk mengeluarkan udara keluar dari dalam kotak. Disisi lain sensor gas juga aktif dan melakukan *pre heating* (pemanasan) untuk melakukan *sensing* terhadap konsentrasi gas yang akan diukur.

Keluaran dari sensor yang berupa *analog output* masuk kedalam STM32F4 *Discovery* dan akan di-convert menjadi *Digital Output* yang nantinya dapat diprogram menjadi satuan ppm. Kemudian data konsentrasi gas yang telah diukur tersebut akan ditampilkan ke dalam LCD 16x4 dan dilakukan proses

penyimpanan data (*data logger*) di dalam modul *SD Card*. Penyimpanan data ini dilakukan selama 5 detik sekali. Data yang tersimpan juga *real time* karena sudah ada penambahan *Real Time Clock (RTC)* DS3231 sehingga data yang tersimpan akan menampilkan tanggal, bulan, tahun, jam, dan detik

Selain itu juga terdapat fitur *sms gateway* yang dirancang untuk memberikan informasi antar *mobile* mengenai hasil pemantaun konsentrasi. Pemberitahuan ini akan dikirim ke *handphone user* apabila konsentrasi gas yang dideteksi oleh empat parameter gas, salah satunya melebihi nilai ambang batas konsentrasi gas masing-masing parameter yang telah diprogram sebelumnya di dalam *STM32F4 Discovery*.

Dibawah ini merupakan perancangan sistem *Monitoring* gas karbon monoksida (CO) berbasis *STM32F4 Discovery*

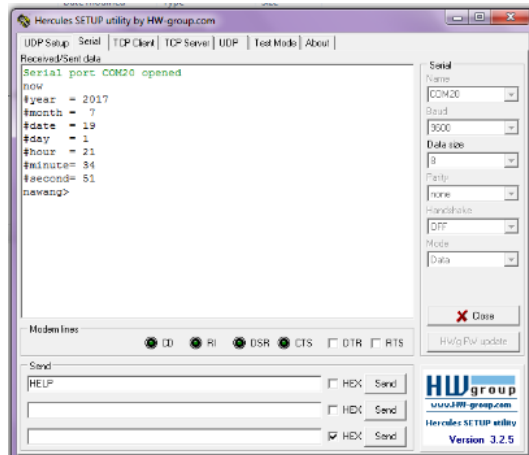


Gambar 4.2 Alat *Monitoring* Gas

4.2 Pengujian *Real Time Clock (RTC)*

Pengujian rangkaian *Real Time Clock* dilakukan dengan memprogram kodingan pada *STM32F4 Discovery*, dimana *RTC* DS3231 berfungsi sebagai timer. *RTC* ini bekerja apabila mendapatkan tegangan sebesar 5V untuk mengaktifkan rangkaian *RTC* DS3231. Pengaturan tanggal, bulan tahun, menit dan jam pada *RTC* dilakukan pengujian melalui *Serial Terminal*. Dibawah

ini adalah pengaturan RTC melalui *serial terminal* pada gambar 4.3



Gambar 4.3 Pengujian data waktu RTC pada *Hyperterminal*

Pada saat pengujian terlihat bahwa RTC DS3231 bisa bekerja dengan baik dengan menampilkan tanggal, bulan tahun, menit dan jam sesuai dengan pengaturan sebelumnya. Data waktu yang sudah sesuai dicantumkan kedalam format *excel* yang akan disimpan ke dalam *SD Card*.

4.3 Pengujian Penyimpanan dalam *SD Card*

Pengujian modul *SD Card* menggunakan pengambilan data yang diperoleh dari empat parameter sensor yakni sensor MQ 7, MQ 136, MG 811 dan MQ 135. Data yang didapatkan dari pembacaan sensor akan disimpan ke dalam *SD Card* setiap 5 detik. Dibawah ini hasil pengujian pembacaan dan penulisan dalam *SD Card* pada gambar 4.4

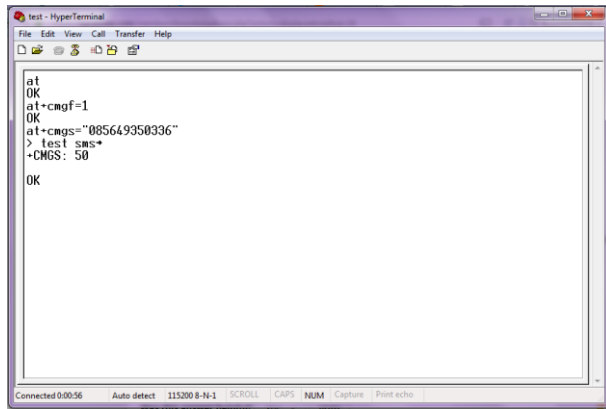
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	10/07/2017	15-49	199	458	635	760														
2	10/07/2017	15-49	199	460	637	749														
3	10/07/2017	15-49	199	460	637	752														
4	10/07/2017	15-49	200	461	638	757														
5	10/07/2017	15-49	199	459	635	759														

Gambar 4.4 Pengujian *SD Card*

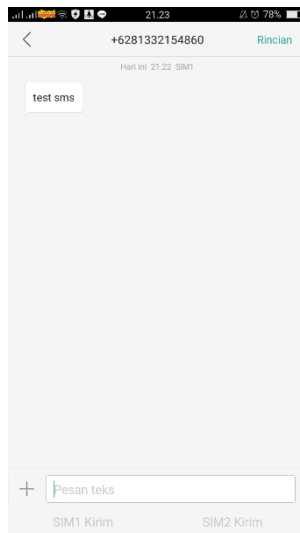
Pada gambar 4.4 diperoleh hasil yang sesuai dengan yang diinginkan yakni data dapat dibaca atau ditulis kedalam memori *SD Card* dari hasil pembacaan sensor MQ 7, MQ 136, MG 811 dan MQ 135.

4.4 Pengujian Modem Wavecom untuk SMS Gateway

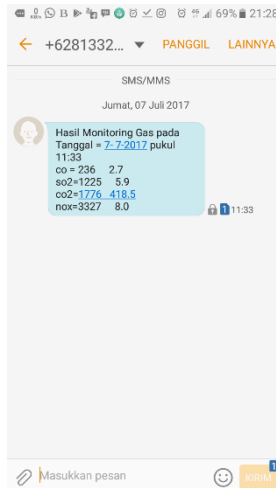
Pengujian sistem *sms gateway* terdiri dari pengiriman dan penerimaan sms. Pengiriman SMS ini hanya diprogram untuk satu nomer *handphone*. Sistem ini bekerja dengan memberikan informasi berupa hasil *Monitoring* gas apabila telah melebihi nilai ambang batas masing-masing parameter sensor. Sehingga pengiriman sms dilakukan sebagai peringatan sekaligus pemberitahuan konsentrasi gas dan kondisinya apabila melebihi ambang nilai ambang batas. Nilai ambang batas karbonmonoksida (CO) berdasarkan PER.13/MEN/X/2011 yaitu 25 ppm. Berikut merupakan hasil pengujian yang telah dilakukan:



Gambar 4.5 *Screenshoot AT Command pada Hyperterminal*



Gambar 4.6 *Screenshoot SMS yang diterima pada HP*



Gambar 4.7 Screenshot SMS yang diterima pada HP melebihi Nilai Ambang Batas

Untuk mengatasi permasalahan agar sms tidak dikirim terus-menerus dalam wilayah ambang batas yang masih sama maka digunakan fungsi *if* dengan kondisi awal seperti berikut ini.

```
//kondisi awal
if (co >= 25)
if(udhkirim==0)
Kirim SMS();
udhkirim=1;
else
udhkirim=0; //(Lampiran)
```

Jika gas CO lebih dari 25 ppm maka akan mengirimkan SMS, jika kondisi SMS sudah terpenuhi maka SMS akan berhenti.

4.5 Pengujian Rangkaian Sensor MQ 7

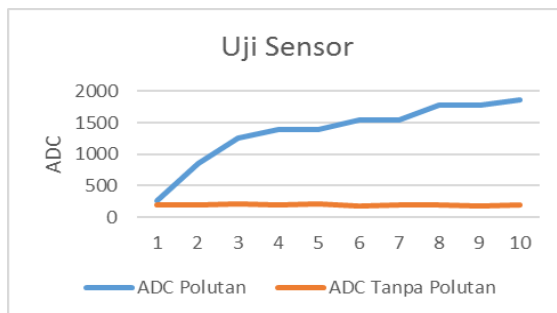
Pada pengujian sensor MQ-7 menggunakan cara pengambilan data yang diperoleh dari uji udara dengan 2 kondisi yaitu udara tanpa polutan dan udara dengan polutan. Sebelum

melakukan uji, sensor MQ 7 harus melakukan *pre heating* (pemanasan) tidak kurang dari 48 jam. Dibawah ini merupakan hasil *Monitoring* uji sensor yang telah dilakukan dengan 2 kondisi yang berbeda.

Tabel 4.1 Uji Sensor Dalam 2 Kondisi

No	Udara tanpa polutan	Udara dengan polutan
	ADC	ADC
1	192	267
2	194	853
3	211	1256
4	200	1392
5	216	1392
6	185	1546
7	198	1546
8	204	1772
9	185	1772
10	190	1870

Dari pengambilan uji diatas, dapat dibuat tabel 4.8 sebagai berikut



Gambar 4.8 Grafik Uji Sensor dengan 2 keadaan

Setelah dilakukan pengambilan data pada uji sensor, didapatkan nilai bahwa perubahan *output* sensor sesuai dengan *variable* (kondisi) yang diujikan. Nilai konsentrasi gas CO yang terdapat pada udara tanpa polutan memiliki output perubahan ppm dalam keadaan stabil. Sebaliknya, nilai konsentrasi gas CO mengalami *output* kenaikan saat kondisi udara dengan polutan.

Setelah mengetahui kondisi sensor sudah bekerja dengan efektif, tahap selanjutnya yakni dilakukan pengambilan data kalibrasi untuk mengetahui tingkat ketidakpastian pengukuran dan data untuk mencari nilai karakteristik statik sensor MQ 7.

Dibawah ini adalah data *Monitoring* yang telah dilakukan pada tabel 4.2

Tabel 4.2 Pengambilan Data *Monitoring*

No	Pembacaan Standar (PPM)	Pembacaan ADC Alat
1	32,30	284
2	29,76	273
3	27,20	262
4	25,80	255
5	22,70	246
6	17,90	217
7	16,46	213
8	14,19	205
9	11,27	196
10	10,94	189
11	6,81	173

Dari data diatas diperoleh grafik perbandingan perbedaan antara pembacaan alat *Monitoring* berupa ADC dengan alat standar berupa *Aeroqual Source*. Sehingga dapat dilakukan perhitungan persamann regresi dengan rumus sebagai berikut:

$$Y = a + bx$$

Dimana

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Keterangan:

x = ADC Pembacaan Alat

y = Pembacaan Standar

a = Konstanta

b = Koefisien regresi

Sehingga dapat dihitung dengan pembuktian matematik untuk mendapatkan persamaan regresinya

$$\begin{aligned} a &= \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \\ &= \frac{(215,33)(587919) - (2513)(52341,22)}{11(587919) - 6315169} \\ &= \frac{126596598,27 - 131533485,}{6467109 - 6315169} \\ &= \frac{-4936887,59}{151940} \end{aligned}$$

$$= -32,492$$

$$\begin{aligned} b &= \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \\ &= \frac{11(52334,34) - (2513)(215,33)}{11(587919) - 6315169} \end{aligned}$$

$$= \frac{575753,42 - 541124,29}{6467109 - 6315169}$$

$$= \frac{34629,13}{151940}$$

$$= 0,2779$$

Sehingga didapatkan persamaan regresi liner seperti dibawah ini:

$$Y = a + bx$$

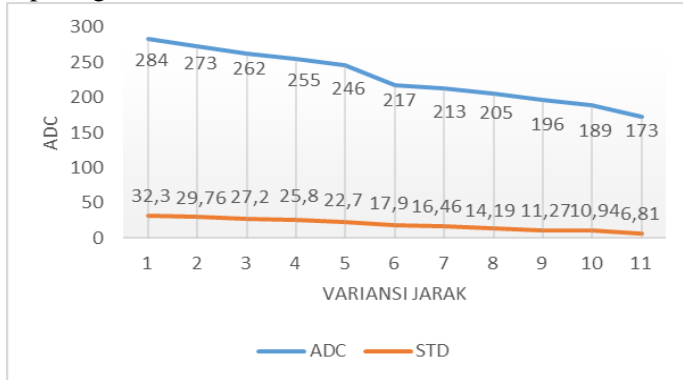
$$= -32,492 + 0,2779x \approx 0,2279x - 32,492$$

Dari perhitungan diatas diperoleh hasil $y = 0,2279x - 32,492$ maka data yang digunakan seperti berikut:

Tabel 4.3 Tabel Konversi Data ADC ke PPM

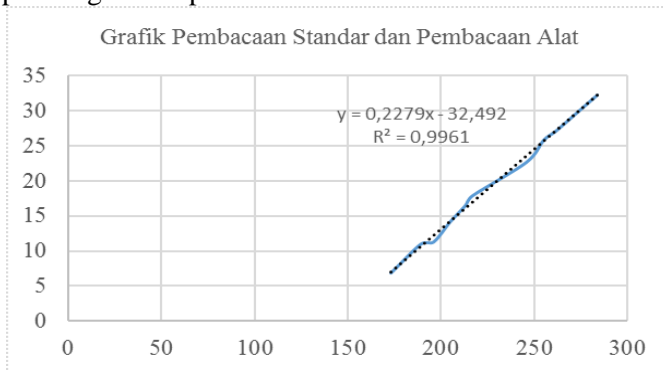
No	Pembacaan Standar	ADC	Fungsi y (Pembacaan Alat)
1	32,30	284	32,23
2	29,76	273	29,72
3	27,20	262	27,22
4	25,80	255	25,62
5	22,70	246	23,57
6	17,90	217	16,96
7	16,46	213	16,05
8	14,19	205	14,23
9	11,27	196	12,18
10	10,94	189	10,58
11	6,81	173	6,93

Dibawah ini gambar grafik yang dihasilkan dari tabel dengan membandingkan pembacaan standar (ppm) dan pembacaan alat (adc) pada gambar 4.9



Gambar 4.9 Grafik Pembacaan Standar dan Pembacaan Alat

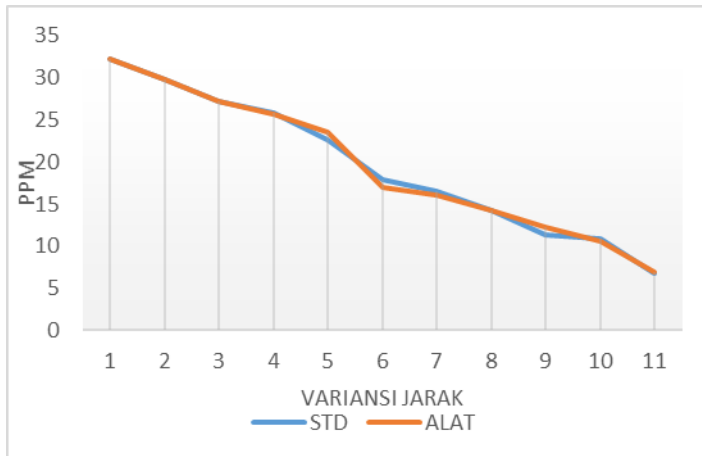
Jika tipe grafik diubah menjadi tipe *scatter*, maka akan didapatkan grafik seperti dibawah ini:



Gambar 4.10 Grafik Pembacaan Standar dan Pembacaan Alat dengan Tipe *Scatter*

Dari grafik diatas menunjukkan bahwa persamaan pengujian dari alat dibandingkat dengan alat ukur yang standar, dimana persamaan yang muncul akan dipakai dalam

programming STM32F4 *Discovery*. Persamaan grafik pengujian ppm yaitu $y = 0,2279x - 32,492$ dimana y sebagai nilai dari pembacaan alat standar dan x sebagai nilai pembacaan alat. Persamaan matematik tersebut menghasilkan data pembacaan alat ppm karbon monoksida (CO) yang sudah mendekati dengan pembacaan alat standar, aeroqual karbon monoksida (CO). Dibawah ini merupakan grafik pengujian pembacaan alat ukur ppm yang telah diberi persamaan matematik. Dibawah ini gambar grafik yang dihasilkan dari pembacaan standar (ppm) dan pembacaan alat (ppm).



Gambar 4.11 Grafik Pembacaan Standar dan Pembacaan Alat dengan persamaan matematik

4.6 Data Spesifikasi Alat

Karakteristik statik digunakan untuk mengukur suatu kondisi yang tidak berubah karena waktu atau hanya berubah secara lambat laun. Sehingga perlu dilakukan perhitungan untuk mengetahui nilai karakteristik dari sensor MQ-7. Berikut ini didapatkan hasil dari pengujian spesifikasi alat melalui data karakteristik statik, menghasilkan data pada tabel 4.4:

Tabel 4.4 Data Spesifikasi Alat

No	Pembacaan STD	Pembacaan Alat	(STD- Alat)/STD	Non- Linearitas
1	32,30	32,23	0,002	0,00
2	29,76	29,72	0,001	0,02
3	27,20	27,22	0,001	0,06
4	25,80	25,62	0,007	-0,16
5	22,70	23,57	0,038	0,32
6	17,90	16,96	0,052	-0,42
7	16,46	16,05	0,025	-0,20
8	14,19	14,23	0,003	-0,05
9	11,27	12,18	0,080	0,75
10	10,94	10,58	0,033	-0,22
11	6,81	6,93	0,018	0,00
Jumlah	215,53	215,30	0,16	0,09
Rata-Rata	19,58	19,57	0,01	0,01

Sehingga dihasilkan nilai:

- Range* : 6,93-32,23 ppm
- Span* : 25,3
- Resolusi : 0,01
- Non-linieritas : 2,96%
- Akurasi : 84%
- Kesalahan (*Error*) : 16%
-

Berikut ini hasil perhitungan nilai karakteristik statik sensor MQ 7 berdasarkan data pada tabel 4.4:

- Non – Linieritas ($N(I) = O(I) - (KI + a)$)

*(Berdasarkan data naik)

Non – Linieritas maksimum per unit

$$= \frac{\hat{N}}{O_{\max} - O_{\min}} \times 100\%$$

Dimana:

$$K \text{ (sensitifitas)} = \frac{\Delta O}{\Delta I} = \frac{32,23 - 6,93}{32,30 - 6,81} = 0,99$$

$$a \text{ (zero bias)} = O_{min} - KI_{min}$$

$$a = 6,93 - (0,99 \times 6,81)$$

$$a = 0,18$$

$$N \text{ (Non linearitas maksimum)} = 0,75$$

Sehingga:

Non – Linieritas maks. Per unit

$$= \frac{0,75}{32,23 - 6,93} \times 100\%$$

$$\text{Non-linearitas} = 2,96\%$$

b. Akurasi:

$$A = 1 - \Sigma \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%$$

Dengan:

Y_n = Pembacaan standar (I) dan

X_n = Pembacaan alat (O)

$$A = 1 - (0,16) \times 100\%$$

$$A = 0,84 \times 100\%$$

$$A = 84\%$$

c. Error:

$$e = 1 - A$$

$$e = 1 - 0,84$$

$$e = 0,16 \times 100\% = 16\%$$

4.7 Data Hasil Kalibrasi

Berikut ini merupakan hasil pengukuran kalibrasi untuk mencari nilai ketidakpastian alat ukur

Tabel 4.5 Data Kalibrasi

NO	STANDAR	ALAT	KOREKSI(Y)	Yreg	RESIDU (R)	SR
1	32,30	32,23	0,07	0,0531	0,0153	0,0002
2	29,76	29,72	0,04	0,0430	-0,0077	0,0001
3	27,20	27,22	-0,02	0,0329	-0,0507	0,0026
4	25,80	25,62	0,18	0,0273	0,1502	0,0226
5	22,70	23,57	-0,87	0,0150	-0,8864	0,7858
6	17,90	16,96	0,94	-0,0040	0,9417	0,8867
7	16,46	16,05	0,41	-0,0097	0,4190	0,1755
8	14,19	14,23	-0,04	-0,0187	-0,0188	0,0004
9	11,27	12,18	-0,91	-0,0302	-0,8762	0,7677
10	10,94	10,58	0,36	-0,0315	0,3904	0,1524
11	6,81	6,93	-0,12	-0,0479	-0,0768	0,0059
Jumlah	215,33	215,30	0,03		SSR=>	2,7998
Rata-Rata	19,58	19,57	0,003			

Berikut merupakan perhitungan ketidakpastian alat ukur berdasarkan tabel 4.3

- a. Nilai Ketidakpastian tipe A:

Standar deviasi :

$$\sigma = \frac{\sqrt{\sum(y_i - \bar{y})^2}}{n - 1}$$

Sehingga nilai ketidakpastian tipe A adalah :

$$U_{A1} = \frac{\sigma}{\sqrt{n}}$$

$$U_{A1} = 0$$

Sehingga nilai ketidakpastian regresi U_{a2} adalah

$$U_{A2} = \sqrt{\frac{SSR}{n-v}}$$

Dimana :

v = variabel bebas = 2

SSR (*Sum Square Residual*) = $\Sigma \text{SR}(\text{Square Residual})$

$\text{SR} = R^2$ (*Residu*)

Y_i (Nilai koreksi) = Pemb. standar (t_i) – Pemb. alat (x_i)

$$Y_{\text{reg}} = a + (b \times t_i)$$

$$a = \bar{y}_1 - (b \times \bar{t}_1)$$

$$b = \frac{n \cdot \sum t_i y_i - \sum y \cdot \sum t_i}{n \cdot \sum t_i^2 - (\sum t_i)^2};$$

t_i = Pemb. standar,

y_i = Nilai koreksi, n = Jumlah data

$$b = \frac{(11 \times 3,43) - (0,03 \times 215,33)}{(11 \times 4935,49) - (215,33)^2}$$

$$b = 0,004$$

Sehingga nilai :

$$a = 0,003 - (0,004 \times 19,58)$$

$$a = -0,075$$

Jadi persamaan regresi menjadi

$$Y_{\text{reg}} = (-0,075) + (t_i \times 0,004)$$

Yang menghasilkan nilai $\text{SSR} = 2,7998$

$$U_{a2} = \sqrt{\frac{2,7998}{11-2}}$$

$$U_{a2} = 0,558$$

b. Nilai ketidakpastian tipe B

Pada ketidakpastian tipe B ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi (U_{B1}) dan ketidakpastian alat standar *multimeter* (U_{B2}). Berikut ini adalah perhitungan ketidakpastian tipe B :

$$U_{B1} = \frac{a}{\sqrt{3}}$$

$$= \frac{-0,075}{\sqrt{3}}$$

$$= -0,043$$

$$U_{B2} = \frac{\frac{1}{2} \times \text{Resolusi}}{\sqrt{3}}$$

$$= \frac{\frac{1}{2} \times 0,01}{\sqrt{3}}$$

$$= 0,003$$

c. Nilai ketidakpastian kombinasi U_c :

$$U_c = \sqrt{U_{A1}^2 + U_{A2}^2 + U_{B1}^2 + U_{B2}^2}$$

$$U_c = \sqrt{0^2 + 0,558^2 + (-0,043)^2 + 0,003^2}$$

$$U_c = 0,589$$

Dengan kondisi V atau derajat kebebasan dari kedua tipe ketidakpastian, sebagai berikut :

$V = n-1$, sehingga :

$$V1 = 10; V2 = 10; V3 = 0,5; V4 = 0,5$$

Dengan nilai V_{eff} (Nilai derajat kebebasan efektif)

$$\text{sebagai } V_{eff} = \frac{(U_c)^4}{\sum (U_i)^4 / v_i}$$

$$V_{eff} = \frac{0,589^4}{\frac{(0)^4}{10} + \frac{(0,558)^4}{10} + \frac{(-0,043)^4}{0,5} + \frac{(0,003)^4}{0,5}}$$

$$V_{eff} = 12,40$$

Sehingga untuk mencari nilai faktor cakupan (k) digunakan rumus interpolasi terdekat.

$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1}$$

$$\frac{12,40-12}{13-12} = \frac{y-2,179}{2,160-2,179}$$

$$\frac{0,40}{1} = \frac{y-2,179}{-0,019}$$

$$-0,0076 = y - 2,179$$

$$-0,0076 + 2,179 = y$$

$$2,171 = y$$

Oleh karena itu, hasil nilai ketidakpastian diperluang sebesar :

$$U_{exp} = k \times U_c$$

$$U_{exp} = 2,171 \times 0,589 = 1,27$$

Sehingga diperoleh hasil pengukuran seperti dibawah ini:

$$x = \bar{x} \pm U_{exp}$$

$$x = 32,23 \pm 1,27 \text{ (ppm)}$$

$$CL = 95\%$$

$$k = 2,171$$

4.8 Pembahasan

Sistem *Monitoring* kadar karbon monoksida (CO) ini terdiri dari satu *variabel* yang diukur yaitu ppm. Pengukuran kadar karbon monoksida (CO) dilakukan oleh sensor gas MQ-7. Hasil pengukuran ditampilkan pada LCD 16x4 dan disimpan ke dalam *sd card* dan di komunikasikan melalui *sms gateway*.

Prinsip kerja dari alat ukur kadar karbon monoksida (CO) ini yang pertama adalah *sensing element* yang dilakukan oleh sensor gas MQ-7. Output dari sensor gas MQ-7 adalah *analog output*. Karena dibutuhkan sinyal *digital*, maka pada *signal conditioning element* diberi komparator yang berfungsi untuk mengkondisikan sinyal agar outputan *high* atau *low* (*digital output*). Pada *signal processing element*, sinyal *digital* diolah dengan STM32F4 *Discovery*. Output dari STM32F4 *Discovery* sudah berupa data visual yang dapat dilihat melalui LCD.

Dilakukannya pengujian alat dalam rancang bangun sistem *Monitoring* emisi gas buang ini khususnya pada pengukuran gas karbon monoksida(CO), agar dapat mengetahui alat dapat bekerja sesuai dengan perancangan.

Pengujian pada RTC, *SD Card* dan *SMS gateway* juga telah dilakukan sehingga dapat mengetahui sistem *Monitoring* dapat bekerja dengan baik. RTC berjalan secara *real time*. *SD Card* dapat menyimpan data dan *sms gateway* dapat mengirimkan pesan informasi hasil *Monitoring* ketika sudah mencapai nilai ambang batas parameter gas.

Pengujian pada sensor gas MQ-7 dilakukan dengan membandingkan hasil pembacaan alat berupa adc menggunakan 2 kondisi (*variable*) yakni pembacaan saat sensor gas tanpa gas polutan dan pembacaan saat terkena gas polutan. Dari gambar 4.8

diperoleh grafik yang menunjukkan bahwa nilai adc pada alat stabil jika tidak terkena gas polutan dan mengalami *output* kenaikan saat kondisi udara dengan polutan.

Pengujian terhadap alat ukur seluruhnya dilakukan dengan membandingkan nilai konsentrasi gas dari pembacaan alat standar (ppm) dengan pembacaan alat berupa nilai adc yang telah diubah menjadi ppm menggunakan persamaan regresi.

Pengambilan data pada pengujian ini sebanyak 11 data. Setelah memperoleh data tersebut, maka dapat diolah untuk mencari data spesifikasi alat untuk mengetahui karakteristik statik dan data kalibrasi untuk mengetahui hasil pengukuran, *uncertainty* (U_{exp}), *confidence level* (CL), faktor cakupan (k).

Pada tabel 4.4 diperoleh data spesifikasi alat sehingga menghasilkan nilai *range* pengukuran yaitu 6,93-32,23 ppm, *span* pengukuran sebesar 25,3, resolusi sebesar 0,01, non-linearitas sebesar 2,96%, akurasi sebesar 84%, kesalahan (*error*) sebesar 16%. Pada tabel 4.5 diperoleh data kalibrasi dengan hasil pengukuran $x = 32,23 \pm 1,27$ (ppm), *confidence level* (CL) 95%, dan faktor cakupan (k) sebesar 2,171

Dari analisis tersebut, Sistem *Monitoring Gas* ini dapat digunakan karena hasil pengukuran yang sesuai dengan standar yang berlaku.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pada hasil penelitian tugas akhir yang sudah dilakukan dapat disimpulkan sebagai berikut:

1. Telah dibuat rancang bangun sistem monitoring konsentrasi gas karbon monoksida (CO) sebagai emisi gas buang menggunakan mikrokontroller STM32F4 *Discovery* dengan spesifikasi alat sebagai berikut : *range* sebesar 6,93-32,23 ppm, *span* sebesar 25,3, resolusi sebesar 0,01, non-linieritas sebesar 2,96%, akurasi sebesar 84%, dan kesalahan (*error*) sebesar 16%
2. Dari hasil kalibrasi yang telah dilakukan, menyatakan bahwa nilai hasil pengukuran $x = 32,23 \pm 1,27$ (ppm), *confidence level* (CL) 95%, dan faktor cakupan (k) sebesar $\pm 2,171$
3. Penggunaan data hasil monitoring digunakan dalam penyimpanan pada *SD Card* dan komunikasi antar *mobile* menggunakan *modem wavecom* secara *real time*.

5.2 Saran

Saran yang diberikan untuk dilakukan penelitian selanjutnya yaitu:

1. Dibutuhkan sampel gas uji yang dapat mengatur *range ppm* dengan baik untuk proses kalibrasi, sehingga hasil dari rancang bangun sistem monitoring ini dapat lebih akurat
2. Alat ini dapat disempurnakan dengan penambahan RTC Eksternal sebagai *Real Time Clock* dalam pengambilan *Data Logger* menggunakan *SD Card* serta komunikasi antar *mobile* dengan fitur *SMS Gateway*.

Halaman Ini Sengaja Dikosongkan

DAFTAR PUSTAKA

- [1] Sentra Informasi Keracunan Nasional. 2010. Sentra Informasi Keracunan Nasional Carbon Monoxide. Jakarta: Badan Pengawasan Obat dan Makanan
- [2] Kementerian Lingkungan Hidup. 2013. Evaluasi kualitas udara perkotaan. Jakarta: Langit biru
- [3] Sentra Informasi Keracunan Nasional. 2010. Carbon Monoxide. Jakarta: Badan Pengawasan Obat dan Makanan
- [4] Faisal, Yunus. 2012 Dampak asap kebakaran hutan pada pernafasan. Surakarta: Cermin Dunia Kedokteran
- [5] Fardiaz, Srikandi. 1992. Polusi Air dan Udara. Kansius: Yogyakarta
- [6] EMD, Armin. 2000. Carbon Monoxide Poisoning. England: Engl J Med
- [7] Anonim. 2011. Karbon Monoksida. <http://digilib.itb.ac.id/files/disk1/560/jbptitbpp-gdl-asepsofyan-27951-3-2001ts-2.pdf> (Diakses 24 Mei 2017, 10.00 WIB)
- [8] Anggraeni NIS. 2009. Pengaruh lama paparan asap knalpot dengan kadar CO 1800 ppm terhadap gambaran histopatologis jantung pada tikus wistar. Semarang: Universitas Diponegoro
- [9] Wardhana, Wisnu Arya. 2001. Dampak Pencemaran Lingkungan. ANDI: Yogyakarta
- [10] Mulya, Araya. 2004. Keracunan Yang Disebabkan Gas Karbon Monoksida. Jakarta: Badan Pengawas Obat dan Makanan
- [11] Liu, Chang-Ting. 2012. Oxygenase-1 System, Inflammation And Ventilator-Induced Lung Injury. European Journal of Pharmacology
- [12] World Health Organization. 2000. "Air quality guidelines Edisi ke-2". Copenhagen: World Health Organization
- [13] Admaja, Bkti. 2011. PER 13 MEN X 2011 NAB Faktor Fisika dan kimia di tempat kerja Udara. . <https://id.scribd.com/doc/72997827/PER-13-MEN-X->

- 2011-NAB-Faktor-Fisika-dan-kimia-di-tempat-kerja Udara.pdf (Diakses 25 Mei 2017, 20.20 WIB)
- [14] Fardiaz, Srikandi. 1992. *Polusi Air Dan Udara*. Kanisus: Yogyakarta.
- [15] Ika, Nur. 2013. Tanda dan Gejala Keracunan Karbon Monoksida. http://eprints.undip.ac.id/13517/1/Nur_Ika_Setyowati_A.pdf (Diakses 26 Mei 2017, 15.00 WIB)
- [16] *Datasheet MQ-7. Toxic Gas Sensor MQ-7 Manual version. 3.* <https://id.scribd.com/document/324297039/MQ-7-Ver1-3-Manual> (Diakses 26 Mei 2017, 15.30 WIB)
- [17] Ardianto, A. 2010. Pemanfaatan Mikrokontroler ATMega8535 dan Sensor PIR sebagai Pengendali Alat Pengering Tangan. UNS, Surakarta.
- [18] Anonim. 2012. ST32F4 Discovery. http://sir.stikom.edu/1701/4/BAB_II.pdf (Diakses 26 Mei 2017, 16.00 WIB)
- [19] *Datasheet STM32F4 Discovery. User Manual Discovery kit with STM32F407VG.* http://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf (Diakses 26 Mei 2017, 16.15 WIB)
- [20] *Datasheet LCD 16x4. Manual ERM1604-1 Series LCD 16x4.* http://www.buydisplay.com/download/manual/ERM1604-1_Series_Datasheet.pdf (Diakses 26 Mei 2017, 17.00 WIB)
- [21] *Datasheet RTC DS3231. Extremely Accurate DS2313.* <https://web.wpi.edu/Pubs/E-project/Available/E-project-010908-124414/unrestricted/DS3231-DS3231S.pdf> (Diakses 26 Mei 2017, 21.00 WIB)
- [22] Anonim. 2011. Modul Mikro SD Card. <https://www.tindie.com/products/mmm999/micro-sd-card-reader-module-for-arduino/> (Diakses 27 Mei 2017, 10.00 WIB)
- [23] Sonoku. 2011. Data Logger Bagian-2. <http://sonoku.com> (Diakses 27 Mei 2017, 11.00)

- [24] Anonim. 2012. Serial RS232. <http://repository.usu.ac.id/bitstream/handle/123456789/42752/Chapter%20II.pdf;jsessionid=8B937B36E42975C825EE710A703ACA83?sequence=3>
- [25] Sugiono, Djoko. 2014. Komunikasi Data Melalui RS232. <http://www.vedcmalang.com/pppstkboemlg/index.php/menutama/listrik-electro/1054-ds1> (Diakses 27 Mei 2017, 12.10 WIB)
- [26] Lamudi. 2014. Pengertian *Exhaust Fan* dan Cara Memilihnya. <http://www.lamudi.co.id/journal/pengertian-exhaust-fan-dan-cara-memilihnya/> (Diakses 27 Mei 2017, 13.00 WIB)
- [27] Irawan, Irfan. 2012. Perbandingan Power Supply Konvensional dengan Power Supply Switching. <http://www.almuhibbin.com/2012/05/perbandingan-power-supply-konvensional.html> (Diakses 27 Mei 2017, 14.00 WIB)
- [28] Kho, Dickson. 2017. Pengertian MCB (Miniature Circuit Breaker) dan Prinsip Kerjanya. <http://teknikelektronika.com/pengertian-mcb-miniature-circuit-breaker-prinsip-kerja-mcb/> (Diakses 27 Mei 2017, 15.00 WIB)
- [29] Indraharja. 2012. Pengertian IC LM7805. <http://indraharja.com/pengertian-iclm7805/> (Diakses 27 Mei 2017, 15.30 WIB)
- [30] Atmaja, Bagus. 2015. Rancang Bangun STM32 dengan ChibiOs. <http://www.bagustris.tk/2015/04/rancang-bangun-modul-stm32-dengan.html> (Diakses 27 Mei 2017, 16.00 WIB)
- [31] Pamungkas, Haryo. 2012. Desain dan implementasi perangkat lunak pada sistem mikrokontroler berbasis ChibiOs/RT (studi kasus pengontrolan motor dc). http://www.elektro.undip.ac.id/el_kpta/wp-content/uploads/2012/05/L2F006047_MTA.pdf (Diakses 28 Mei 2017, 13.50 WIB)

- [32] Fuadi, Taufik. 2013. *Qt Creator*. <http://informatika.unsyiah.ac.id/irvanizam/teaching/prog2/INF103-08.pdf> (Diakses 28 Mei 2017, 14.27 WIB)
- [33] Anonim. STM32 ST-LINK Utility Software Discription. http://www.st.com/content/ccc/resource/technical/document/user_manual/e6/10/d8/80/d6/1d/4a/f2/CD00262073.pdf/files/CD00262073.pdf/jcr:content/translations/en.CD00262073.pdf (Diakses 30 Mei 2017, 13.02 wib)
- [34] Laboratorium Pengukuran Fisis. 2013. Modul Teknik Pengukuran dan Kalibrasi. Teknik Fisika, FFI-ITS, Surabaya.

LAMPIRAN A

(*LISTING PROGRAM PADA CODE VISION AVR*)

/******

ChibiOS/RT - Copyright (C) 2006-2013 Giovanni Di Sirio

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

*****/

//main.c

#include "ch.h"

#include "hal.h"

#include "ta_adc.h"

#include "ta_lcd.h"

#include "ta_utama.h"

#include "ta_shell.h"

#include "ta_i2c.h"

#include "ta_rtc.h"

#include "ta_mmc.h"

#include "ta_uart.h"

```

int main(void) {
    halInit();
    chSysInit();
    Adc_Init();
    Lcd_Init();
    Lcd_Clear();
    Ds1307_Init();
    Sms_Text();
    Mmc_Init();
    Shell_Init();
    Run_Init();
    while (TRUE) {
        Shell_Run();
    }
}

```

//adc.c

```
#include "ta_adc.h"
```

```

static adcsample_t samples[ADC_GRP1_NUM_CHANNELS *
    ADC_GRP1_BUF_DEPTH];
adcsample_t adc0,adc1,adc2,adc3;
uint32_t sum_adc0,sum_adc1,sum_adc2,sum_adc3;
void adccb(ADCDriver *adcp, adcsample_t *buffer, size_t n){
    (void) buffer; (void) n;
    int i;
    if (adcp->state == ADC_COMPLETE) {
        sum_adc0=0;
        sum_adc1=0;
        sum_adc2=0;
        sum_adc3=0;
        for(i=0;i<ADC_GRP1_BUF_DEPTH;i++){
            sum_adc0=sum_adc0+samples[0+(i*ADC_GRP1_NUM_
                CHANNELS)];
            sum_adc1=sum_adc1+samples[1+(i*ADC_GRP1_NUM_
                CHANNELS)];

```

```

        sum_adc2=sum_adc2+samples[2+(i*ADC_GRP1_NUM_CHANNELS)];
        sum_adc3=sum_adc3+samples[3+(i*ADC_GRP1_NUM_CHANNELS)];
    }
    adc0=sum_adc0/ADC_GRP1_BUF_DEPTH;
    adc1=sum_adc1/ADC_GRP1_BUF_DEPTH;
    adc2=sum_adc2/ADC_GRP1_BUF_DEPTH;
    adc3=sum_adc3/ADC_GRP1_BUF_DEPTH;
}
}
static const ADCConversionGroup adcgrpcf = {
    FALSE,
    ADC_GRP1_NUM_CHANNELS,
    adccb,
    NULL,
    /* HW dependent part */
    0,
    ADC_CR2_SWSTART,
    0,
    ADC_SMPR2_SMP_AN0(ADC_SAMPLE_112)
        ADC_SMPR2_SMP_AN1(ADC_SAMPLE_112)
        ADC_SMPR2_SMP_AN2(ADC_SAMPLE_112)
        ADC_SMPR2_SMP_AN3(ADC_SAMPLE_112),
    ADC_SQR1_NUM_CH(ADC_GRP1_NUM_CHANNELS),
    0,
    ADC_SQR3_SQ1_N(ADC_CHANNEL_IN0)
        ADC_SQR3_SQ2_N(ADC_CHANNEL_IN1)
        ADC_SQR3_SQ3_N(ADC_CHANNEL_IN2)
        ADC_SQR3_SQ4_N(ADC_CHANNEL_IN3)
};
static THD_WORKING_AREA(wa_adcThread, 128);
static THD_FUNCTION(adcThread, arg) {
    (void)arg;
    chRegSetThreadName("ADC Run");
    while (TRUE) {

```

```

    chThdSleepMilliseconds(100);
    palSetPad(GPIOD, 12); /* Yellow. */
    adcStartConversion(&ADCD1,      &adcgrpcfg,      samples,
        ADC_GRP1_BUF_DEPTH);
    chThdSleepMilliseconds(100);
    palClearPad(GPIOD, 12); /* Yellow. */
}
}

void Adc_Init(){
    palSetPadMode(GPIOA,0,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,1,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,2,PAL_MODE_INPUT_ANALOG);
    palSetPadMode(GPIOA,3,PAL_MODE_INPUT_ANALOG);
    adcStart(&ADCD1, NULL);
    adcSTM32EnableTSVREFE();
    palSetPadMode(GPIOD,12,PAL_MODE_OUTPUT_PUS
        HPULL);
    chThdCreateStatic(wa_adcThread,      sizeof(wa_adcThread),
        NORMALPRIO, adcThread, NULL);
}

```

//adc.h

```

#ifndef TA_ADC_H
#define TA_ADC_H

#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <stdarg.h>
#include <stdlib.h>
#include <math.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"

```

```

#ifndef CHPRINTF_USE_FLOAT
#define CHPRINTF_USE_FLOAT FALSE
#endif

#define MAX_FILLER 16
#define FLOAT_PRECISION 100
#define ADC_GRP1_NUM_CHANNELS 4
#define ADC_GRP1_BUF_DEPTH 100
void Adc_Init(void);
#endif

//i2c.c
#include "ta_i2c.h"

static const I2CConfig i2cconfig= {
    OPMODE_I2C,
    400000,
    FAST_DUTY_CYCLE_2,
};

uint8_t readByteI2C(uint8_t addr){
    uint8_t data;
    i2cAcquireBus(&I2CD1);
    (void)
        i2cMasterReceiveTimeout(&I2CD1,addr,&data,1,TIME_I
            NFINITE);
    i2cReleaseBus(&I2CD1);
    return data;
}

void writeByteI2C(uint8_t addr, uint8_t reg, uint8_t val){
    uint8_t cmd[] = {reg, val};
    i2cAcquireBus(&I2CD1);
    (void) i2cMasterTransmitTimeout(&I2CD1, addr, cmd, 2,
        NULL, 0, TIME_INFINITE);
    i2cReleaseBus(&I2CD1);
}

void I2c_Init(void){

```

```

    palSetPadMode(GPIOB,8,PAL_MODE_ALTERNATE(4) |
        PAL_STM32_OTYPE_OPENDRAIN);
    palSetPadMode(GPIOB,9,PAL_MODE_ALTERNATE(4) |
        PAL_STM32_OTYPE_OPENDRAIN);
    i2cStart(&I2CD1, &i2cconfig);
}

```

//i2c.h

```

#ifndef TA_I2C_H
#define TA_I2C_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
uint8_t readByteI2C(uint8_t addr);
void writeByteI2C(uint8_t addr, uint8_t reg, uint8_t val);
void I2c_Init(void);
#endif

```

//lcd.c

```

#include "ta_lcd.h"

```

```

LcdStream myLCD;
static msg_t put(void *ip, uint8_t chr) {
    (void)ip;
    Lcd_Write_Data(chr);
    return MSG_OK;
}
static const struct LcdStreamVMT vmt = {NULL, NULL, put,
    NULL};
void lsObjectInit(LcdStream *msp) {

```



```

    msp->vmt = &vmt;
}
void Lcd_Pin_Dir(void){
    palSetPadMode(LCD_PORT_CRTL,LCD_PIN_RS,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_CRTL,LCD_PIN_EN,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D4,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D5,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D6,LCD_
PORT_MODE);
    palSetPadMode(LCD_PORT_DATA,LCD_PIN_D7,LCD_
PORT_MODE);
}
void Lcd_Write_Data(uint8_t chr){
    palWritePort(LCD_PORT_DATA,(chr & 0xf0));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_RS);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(10);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_RS);
    chThdSleepMilliseconds(10);

    palWritePort(LCD_PORT_DATA,((chr & 0x0f)<<4));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_RS);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(10);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_RS);
    chThdSleepMilliseconds(10);
}
void Lcd_Write_Command(uint8_t cmd){
    palWritePort(LCD_PORT_DATA,(cmd & 0xf0));
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);

```

```

chThdSleepMilliseconds(10);
palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);

palWritePort(LCD_PORT_DATA,((cmd & 0x0f)<<4));
palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);
palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
chThdSleepMilliseconds(10);
}
void Lcd_Cursor(uint8_t column, uint8_t line){
    uint8_t position = 0x00;
    if(column>=TLCD_MAXX) column=0;
    if(line>=TLCD_MAXY) line=0;
    switch(line)
    {
        case 0: position = LCD_LINE0_DDRAMADDR+column;
                break;
        case 1: position = LCD_LINE1_DDRAMADDR+column;
                break;
        case 2: position = LCD_LINE2_DDRAMADDR+column;
                break;
        case 3: position = LCD_LINE3_DDRAMADDR+column;
                break;
    }
    Lcd_Write_Command(1<<LCD_DDRAM | position);
}
void Lcd_Init(void){
    lsObjectInit(&myLCD);
    Lcd_Pin_Dir();
    chThdSleepMilliseconds(500);
    palWritePort(LCD_PORT_CRTL,0x00);
    palWritePort(LCD_PORT_DATA,0x00);

    palSetPad(LCD_PORT_DATA,LCD_PIN_D5);
    palSetPad(LCD_PORT_CRTL,LCD_PIN_EN);

```

```

    chThdSleepMilliseconds(40);
    palClearPad(LCD_PORT_CRTL,LCD_PIN_EN);
    chThdSleepMilliseconds(40);

    Lcd_Write_Command(0x28);
    chThdSleepMilliseconds(10);
    Lcd_Write_Command(0x0c);
    chThdSleepMilliseconds(10);
}
void Lcd_Clear (void){
    Lcd_Write_Command(0x01);
    chThdSleepMilliseconds(10);
}
void Lcd_Example(){
    Lcd_Clear();
    Lcd_Cursor(0,0);
    chprintf((BaseSequentialStream *)&myLCD,"A-LCD");
    Lcd_Cursor(0,1);
    chprintf((BaseSequentialStream *)&myLCD,"Works");
    Lcd_Cursor(0,2);
    chprintf((BaseSequentialStream *)&myLCD,"horee");
    Lcd_Cursor(0,3);
    chprintf((BaseSequentialStream *)&myLCD,"yeee");
}

```

//lcd.h

```

#ifndef TA_LCD_H
#define TA_LCD_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"

```

```

#include "chstreams.h"
#define _lcd_stream_data _base_sequential_stream_data
#define LCD_PIN_RS 0
#define LCD_PIN_EN 1
#define LCD_PORT_CRTL GPIOE
#define LCD_PIN_D4 4
#define LCD_PIN_D5 5
#define LCD_PIN_D6 6
#define LCD_PIN_D7 7
#define LCD_PORT_DATA GPIOE
#define                                LCD_PORT_MODE
                                PAL_MODE_OUTPUT_PUSHPULL
#define TLCD_MAXX            16 // max x-Position (0...15)
#define TLCD_MAXY            4 // max y-Position (0...1)
#define LCD_DDRAM            7
#define LCD_LINE0_DDRAMADDR            0x00
#define LCD_LINE1_DDRAMADDR            0x40
#define LCD_LINE2_DDRAMADDR            0x10
#define LCD_LINE3_DDRAMADDR            0x50

struct LcdStreamVMT {
    _base_sequential_stream_methods
};
typedef struct {
    const struct LcdStreamVMT *vmt;
    _base_sequential_stream_data
} LcdStream;

#ifdef __cplusplus
extern "C" {
#endif
void lsObjectInit(LcdStream *msp);
#ifdef __cplusplus
}
#endif

```

```

void Lcd_Pin_Dir(void);
void Lcd_Write_Command(uint8_t cmd);
void Lcd_Write_Data(uint8_t chr);
void Lcd_Init(void);
void Lcd_Cursor(uint8_t column, uint8_t line);
void Lcd_Clear (void);
void Lcd_Example (void);
#endif // LIB_LCD_H

```

//mmc.c

```
#include "ta_mmc.h"
```

```

FATFS MMC_FS;
MMCDriver MMCD1;
bool fs_ready = FALSE;
FRESULT err;
uint32_t clusters;
FATFS *fsp;
uint8_t fbuff[1024];
static SPIConfig hs_spicfg = {NULL, GPIOB, 12, 0};
static SPIConfig ls_spicfg = {NULL, GPIOB,
    12,SPI_CR1_BR_2 | SPI_CR1_BR_1};
static MMCCfg mmccfg = {&SPID2, &ls_spicfg,
    &hs_spicfg};
FRESULT scan_files(BaseSequentialStream *chp, char *path) {
    FRESULT res;
    FILINFO fno;
    DIR dir;
    int i;
    char *fn;

    #if _USE_LFN
        fno.lfname = 0;
        fno.lfsize = 0;
    #endif
    res = f_opendir(&dir, path);

```

```

if (res == FR_OK) {
    i = strlen(path);
    for (;;) {
        res = f_readdir(&dir, &fno);
        if (res != FR_OK || fno.fname[0] == 0)
            break;
        if (fno.fname[0] == '.')
            continue;
        fn = fno.fname;
        if (fno.fattrib & AM_DIR) {
            path[i++] = '/';
            strcpy(&path[i], fn);
            res = scan_files(chp, path);
            if (res != FR_OK)
                break;
            path[--i] = 0;
        }
        else {
            chprintf(chp, "%s/%s\r\n", path, fn);
        }
    }
}
return res;
}

void Mmc_Mount(void) {
    if (fs_ready) {
        return;
    }
    if (mmcConnect(&MMCD1)) {
        return;
    }
    err = f_mount(&MMC_FS, "/", 1);
    if (err != FR_OK) {
        mmcDisconnect(&MMCD1);
        fs_ready = FALSE;
        return;
    }
}

```

```

    }
    fs_ready = TRUE;
}
void Mmc_Unmount(void) {
    f_mount(NULL,"/",1);
    mmcDisconnect(&MMCD1);
    fs_ready = FALSE;
}
FRESULT f_append (
    FIL* fp,          /* [OUT] file object to create */
    const char* path  /* [IN]  file name to be opened */
)
{
    FRESULT fr;
    /* Opens an existing file. If not exist, creates a new file. */
    fr = f_open(fp, path, FA_WRITE | FA_OPEN_ALWAYS |
        FA_READ);
    if (fr == FR_OK) {
        /* Seek to end of the file to append data */
        fr = f_lseek(fp, f_size(fp));
        if (fr != FR_OK)
            f_close(fp);
    }
    return fr;
}
void Mmc_Init(){
    palSetPadMode(GPIOB,13,PAL_MODE_ALTERNATE(5) |
        PAL_STM32_OSPEED_HIGHEST); //SCK

    palSetPadMode(GPIOB,12,PAL_MODE_OUTPUT_PUS
        HPULL | PAL_STM32_OSPEED_HIGHEST); //NSS
    palSetPadMode(GPIOC,2,PAL_MODE_ALTERNATE(5));
        //MISO
    palSetPadMode(GPIOC,3,PAL_MODE_ALTERNATE(5) |
        PAL_STM32_OSPEED_HIGHEST); //MOSI
    palSetPad(GPIOB, 12);

```

```

    mmcObjectInit(&MMCD1);
    mmcStart(&MMCD1, &mmccfg);
    chThdSleepMilliseconds(50);
    Mmc_Mount();
}

```

//mmc.h

```

#ifndef TA_MMC_H
#define TA_MMC_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "evtimer.h"
#include "chvt.h"
#include "ff.h"
#include "ffconf.h"
#define buffer_size 16
FRESULT f_append (FIL* fp, const char* path);
FRESULT scan_files(BaseSequentialStream *chp, char *path);
void Mmc_Mount(void);
void Mmc_Unmount(void);
void Mmc_Init(void);
#endif // TA_MMC_H

```

//rtc.c

```

#include "ta_rtc.h"

struct ds1307_t calendar;
static uint8_t rxbuf[DS1307_RX_DEPTH];
static uint8_t txbuf[DS1307_TX_DEPTH];

```



```

static i2cflags_t errors = 0;
uint8_t bcd2Dec ( uint8_t val )
{
    uint8_t res = ((val/16*10) + (val % 16));
    return res;
}
uint8_t dec2Bcd ( uint8_t val )
{
    uint8_t res = ((val/10*16) + (val%10));
    return res;
}
void setDs1307Date ( msg_t *status, systime_t *tmo, struct
    ds1307_t dsData )
{
    txbuf[0] = DS1307_SECONDS_REG;
    txbuf[1] = dec2Bcd( dsData.seconds );
    txbuf[2] = dec2Bcd( dsData.minutes );
    txbuf[3] = dec2Bcd( dsData.hours );
    txbuf[4] = dec2Bcd( dsData.day );
    txbuf[5] = dec2Bcd( dsData.date );
    txbuf[6] = dec2Bcd( dsData.month );
    txbuf[7] = dec2Bcd( dsData.year - 2000);

    i2cAcquireBus ( &I2CD1 );
    *status = i2cMasterTransmitTimeout ( &I2CD1,
        DS1307_ADDRESS, txbuf, DS1307_TX_DEPTH, rxbuf,
        0, *tmo );
    i2cReleaseBus ( &I2CD1 );
}

struct ds1307_t getDs1307Date ( msg_t *status, systime_t *tmo )
{
    struct ds1307_t dsData;
    txbuf[0] = DS1307_SECONDS_REG;
    i2cAcquireBus( &I2CD1 );

```

```

*status    =    i2cMasterTransmitTimeout    (    &I2CD1,
        DS1307_ADDRESS, txbuf, 1,rxbuf, 7, *tmo );
i2cReleaseBus ( &I2CD1 );
if ( *status != MSG_OK )
{
    errors = i2cGetErrors ( &I2CD1 );
}
else
{
    dsData.seconds = bcd2Dec ( rxbuf[0] & 0x7F );
    dsData.minutes = bcd2Dec ( rxbuf[1] );
    dsData.hours   = bcd2Dec ( rxbuf[2] & 0x3F );
    dsData.day     = bcd2Dec ( rxbuf[3] );
    dsData.date    = bcd2Dec ( rxbuf[4] );
    dsData.month   = bcd2Dec ( rxbuf[5] );
    dsData.year    = bcd2Dec ( rxbuf[6] ) + 2000;
}
return dsData;
}
static THD_WORKING_AREA(waRTC, 128);
static THD_FUNCTION(ThdRTC, arg) {
    (void)arg;
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    chRegSetThreadName("RTC Request");
    while (TRUE) {
        calendar = getDs1307Date ( &status, &timeOut );
        palSetPad(GPIOD, 14);    /* Red. */
        chThdSleepMilliseconds(500);
        palClearPad(GPIOD, 14);    /* Red. */
        chThdSleepMilliseconds(500);
    }
}
void Ds1307_Init ( void )
{
    I2c_Init();

```

```

chThdSleepMilliseconds(500);
    palSetPadMode(GPIOD,14,PAL_MODE_OUTPUT_PUS
        HPULL);
chThdCreateStatic(waRTC, sizeof(waRTC), NORMALPRIO,
    ThdRTC, NULL);
}

```

//rtc.h

```

#ifndef TA_RTC_H
#define TA_RTC_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "shell.h"
#include "ta_i2c.h"
#define DS1307_RX_DEPTH 7
#define DS1307_TX_DEPTH 8
#define DS1307_ADDRESS    0x68
#define DS1307_SECONDS_REG    0x00
typedef struct ds1307_t
{
    uint8_t    seconds;
    uint8_t    minutes;
    uint8_t    hours;
    uint8_t    day;
    uint8_t    date;
    uint8_t    month;
    uint16_t    year;
} ds1307;
uint8_t bcd2Dec ( uint8_t val );

```

```

uint8_t dec2Bcd ( uint8_t val );
void Ds1307_Init ( void );
void setDs1307Date ( msg_t *status, systime_t *tmo, struct
    ds1307_t dsData );
struct ds1307_t getDs1307Date ( msg_t *status, systime_t *tmo );
#endif

```

//shell.c

```
#include "ta_shell.h"
```

```

extern uint16_t adc_co,adc_so2,adc_co2,adc_nox;
thread_t *shelltp = NULL;
extern const USBConfig usbcfg;
extern SerialUSBConfig serusbcfg;
extern struct ds1307_t calendar;
extern FATFS MMC_FS;
extern bool fs_ready;
extern uint32_t clusters;
extern FATFS *fsp;
extern uint8_t fbuff[1024];
extern FRESULT err;
SerialUSBDriver SDU1;
static void cmd_mem(BaseSequentialStream *chp, int argc, char
    *argv[]) {
    size_t n, size;
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: mem\r\n");
        return;
    }
    n = chHeapStatus(NULL, &size);
    chprintf(chp, "core free memory : %u bytes\r\n",
        chCoreGetStatusX());
    chprintf(chp, "heap fragments : %u\r\n", n);
    chprintf(chp, "heap free total : %u bytes\r\n", size);
}

```

```

static void cmd_threads(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    static const char *states[] = {CH_STATE_NAMES};
    thread_t *tp;
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: threads\r\n");
        return;
    }
    chprintf(chp, "  addr  stack prio refs  state time\r\n");
    tp = chRegFirstThread();
    do {
        chprintf(chp, "%08lx %08lx %4lu %4lu %9s\r\n",
            (uint32_t)tp, (uint32_t)tp->p_ctx.r13,
            (uint32_t)tp->p_prio, (uint32_t)(tp->p_refs - 1),
            states[tp->p_state]);
        tp = chRegNextThread(tp);
    } while (tp != NULL);
}

static void cmd_now(BaseSequentialStream *chp, int argc, char
    *argv[]) {

    (void)argv;
    if (argc > 0) {
        chprintf(chp, "Usage: now\r\n");
        return;
    }
    chprintf(chp, "#year = %4i\r\n",calendar.year);
    chprintf(chp, "#month = %2i\r\n",calendar.month);
    chprintf(chp, "#date = %2i\r\n",calendar.date);
    chprintf(chp, "#day = %1i\r\n",calendar.day);
    chprintf(chp, "#hour = %2i\r\n",calendar.hours);
    chprintf(chp, "#minute= %2i\r\n",calendar.minutes);
    chprintf(chp, "#second= %2i\r\n",calendar.seconds);
}

```

```

static void cmd_settime(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    if (argc != 3) {
        chprintf(chp, "Usage: settime sec min hr\r\n");
        return;
    }
    calendar.seconds = atoi(argv[0]);
    calendar.minutes = atoi(argv[1]);
    calendar.hours = atoi(argv[2]);
    setDs1307Date( &status, &timeOut, calendar);
    chprintf(chp, "time was set\r\n");
}

static void cmd_setdate(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    msg_t status = MSG_OK;
    systime_t timeOut = MS2ST ( 4 );
    if (argc != 3) {
        chprintf(chp, "Usage: setdate date month year\r\n");
        return;
    }
    calendar.date = atoi(argv[0]);
    calendar.month = atoi(argv[1]);
    calendar.year = atoi(argv[2]);
    setDs1307Date( &status, &timeOut, calendar);
    chprintf(chp, "date was set\r\n");
}

static void cmd_mmctree(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmctree\r\n");
        return;
    }
}

```

```

if (!fs_ready) {
    chprintf(chp, "File System not mounted\r\n");
    return;
}
err = f_getfree("/", &clusters, &fsp);
if (err != FR_OK) {
    chprintf(chp, "FS: f_getfree() failed (%i)\r\n",err);
    return;
}
chprintf(chp,"FS: %lu free clusters, %lu sectors per cluster,
               %lu          bytes          free\r\n",clusters,
               (uint32_t)MMC_FS.csize,clusters          *
               (uint32_t)MMC_FS.csize                    *
               (uint32_t)MMC_SECTOR_SIZE);
fbuff[0] = 0;
scan_files(chp, (char *)fbuff);
}
static void cmd_mmctest(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmctest\r\n");
        return;
    }
    if (!fs_ready) {
        chprintf(chp, "File System not mounted\r\n");
        return;
    }
    FIL FLogFile;
    memset(&FLogFile, 0, sizeof(FIL));
    FRESULT err_file;
    UINT bw;
    char buffer[buffer_size];
    err_file = f_open(&FLogFile, "Test.txt", FA_WRITE |
        FA_OPEN_ALWAYS);

```

```

if (err_file == FR_OK || err_file == FR_EXIST){
    err_file = f_lseek(&FDLogFile, f_size(&FDLogFile));
    if(err_file == FR_OK){
        chsnprintf(buffer,buffer_size,"Aku Jomblo!!!\n\r");
        f_write(&FDLogFile, buffer, strlen(buffer), &bw);
        f_close(&FDLogFile);
        chprintf(chp, "Some text written\r\n");
        return;
    }else{
        chprintf(chp, "Failed to seek file\r\n");
        return;
    }
}
}
}
static void cmd_mmcadc(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void)argv;
    if (argc > 0) {
        chprintf(chp, "mmcadc\r\n");
        return;
    }
    if (!fs_ready) {
        chprintf(chp, "File System not mounted\r\n");
        return;
    }
    Tulis_Adc();
}
static void cmd_dataadc(BaseSequentialStream *chp, int argc,
    char *argv[]) {
    (void) argv;
    if (argc > 0) {
        chprintf(chp, "Usage: dataadc\r\n");
        return;
    }
}

```



```

    }
    {
        chprintf(chp, "adc_co = %4i\r\n",adc_co);
        chprintf(chp, "adc_so2= %4i\r\n",adc_so2);
        chprintf(chp, "adc_co2= %4i\r\n",adc_co2);
        chprintf(chp, "adc_nox= %4i\r\n",adc_nox);
    }
}

static void cmd_testsms(BaseSequentialStream *chp, int argc,
                        char *argv[]) {
    (void) argv;
    if (argc > 0) {
        chprintf(chp, "Usage: testsms\r\n");
        return;
    }
    Sms_Test();
}

static const ShellCommand commands[] = {
    {"mem", cmd_mem},
    {"threads", cmd_threads},
    {"now", cmd_now},
    {"settime", cmd_settime},
    {"setdate", cmd_setdate},
    {"mmctree", cmd_mmctree},
    {"mmctest", cmd_mmctest},
    {"mmcadc", cmd_mmcadc },
    {"dataadc", cmd_dataadc},
    {"testsms", cmd_testsms},
    {NULL, NULL}
};

static const ShellConfig shell_cfg = {
    (BaseSequentialStream *)&SDU1,
    commands
};

void Shell_Init(void){
    sduObjectInit(&SDU1);

```

```

    sduStart(&SDU1, &serusbcfg);
    usbDisconnectBus(serusbcfg.usbp);
    chThdSleepMilliseconds(1000);
    usbStart(serusbcfg.usbp, &usbcfg);
    usbConnectBus(serusbcfg.usbp);
    shellInit();
}
void Shell_Run(void){
    if (!shelltp && (SDU1.config->usbp->state ==
        USB_ACTIVE))
        shelltp = shellCreate(&shell_cfg, SHELL_WA_SIZE,
            NORMALPRIO);
    else if (chThdTerminatedX(shelltp)) {
        chThdRelease(shelltp); /* Recovers memory of the previous
            shell. */
        shelltp = NULL; /* Triggers spawning of a new shell.
            */
    }
    chThdSleepMilliseconds(1000);
}

```

//shell.h

```

#ifndef TA_SHELL_H
#define TA_SHELL_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "shell.h"
#include "ta_usbcfg.h"
#include "ta_utama.h"

```

```

#include "ta_mmc.h"
#include "ta_uart.h"
#define                                SHELL_WA_SIZE
    THD_WORKING_AREA_SIZE(4096)
#define                                TEST_WA_SIZE
    THD_WORKING_AREA_SIZE(256)

void Shell_Init(void);
void Shell_Run(void);
#endif // TA_SHELL_H

```

//uart.c

```

#include "ta_uart.h"

extern struct ds1307_t calendar;
extern uint16_t adc_co,adc_so2,adc_co2,adc_nox;
extern float v_co,v_so2,v_co2,v_nox;
void Uart_Init(void){
    palSetPadMode(GPIOB,11,PAL_MODE_ALTERNATE(7));
    palSetPadMode(GPIOB,10,PAL_MODE_ALTERNATE(7));
    sdStart(&SD3,NULL);
}
void Sms_Text(void){
    Uart_Init();
    chThdSleepMilliseconds(500);
}
void Sms_Test(void){
    chprintf((BaseSequentialStream *)&SD3,"AT+CMGF=1\n");
    chThdSleepMilliseconds(100);
    chprintf((BaseSequentialStream *)&SD3,"AT+CMGS=\"");
    chprintf((BaseSequentialStream *)&SD3,"+6282244105564");
    chprintf((BaseSequentialStream *)&SD3,"\\\"\\n");
    chThdSleepMilliseconds(100);
    chprintf((BaseSequentialStream *)&SD3,"Hasil Monitoring
        Gas pada\n");
}

```

```

chprintf((BaseSequentialStream *)&SD3,"Tanggal =%2i-%2i-
%4i                                     pukul
%2i:%2i\n",calendar.date,calendar.month,calendar.year,cal
endar.hours,calendar.minutes);
chprintf((BaseSequentialStream *)&SD3,"co      =%4i
%7.1f\n",adc_co,v_co);
chprintf((BaseSequentialStream *)&SD3,"so2=%4i
%7.1f\n",adc_so2,v_so2);
chprintf((BaseSequentialStream *)&SD3,"co2=%4i
%7.1f\n",adc_co2,v_co2);
chprintf((BaseSequentialStream *)&SD3,"nox=%4i
%7.1f\n",adc_nox,v_nox) ;
chThdSleepMilliseconds(100);

chSequentialStreamPut((BaseSequentialStream
*)&SD3,0x1A);
chSequentialStreamPut((BaseSequentialStream
*)&SD3,0x0D);
chSequentialStreamPut((BaseSequentialStream
*)&SD3,0x0A);
chprintf((BaseSequentialStream *)&SD3,"\n");
}

```

//uart.h

```

#ifndef TA_UART_H
#define TA_UART_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "memstreams.h"
#include "chstreams.h"
#include "ta_rtc.h"

```

```
#include "ta_utama.h"

void Uart_Init(void);
void Sms_Text(void);
void Sms_Test(void);
#endif // TA_UART_H
```

//utama.c

```
#include "ta_utama.h"

extern LcdStream myLCD;
extern adcsample_t adc0,adc1,adc2,adc3;
uint16_t adc_co,adc_so2,adc_co2,adc_nox;
float v_co,v_so2,v_co2,v_nox,v_in,R_S;
extern struct ds1307_t calendar;
extern SerialUSBDriver SDU1;
extern FATFS MMC_FS;
uint8_t udhkirim=0;
static THD_WORKING_AREA(waBlink, 128);
static THD_FUNCTION(Blink, arg) {

    (void)arg;
    chRegSetThreadName("Blinker");
    while (TRUE) {
        palSetPad(GPIOD, 13);    /* Orange. */
        chThdSleepMilliseconds(500);
        palClearPad(GPIOD, 13);  /* Orange. */
        chThdSleepMilliseconds(500);
    }
}

static THD_WORKING_AREA(waADCLCD, 128);
static THD_FUNCTION(ADCLCD, arg) {

    (void)arg;
    chRegSetThreadName("ADC LCD");
    while (TRUE) {
```

```

    Hasil_Adc();
}
}
static THD_WORKING_AREA(waRECORD, 1024);
static THD_FUNCTION(RECORD, arg) {

    (void)arg;
    chRegSetThreadName("RECORD");
    while (TRUE) {
        Tulis_Adc();
        palClearPad(GPIOD, 15);
        chThdSleepMilliseconds(5000);
        if ((v_co >= 25) || (v_so2 >= 20) || (v_co2 >= 600) || (v_nox
            >= 4)){
            if(udhkirim==0){
                Sms_Test();
                udhkirim=1;
            }
        }
        else{
            udhkirim=0;
        }
    }
}

void Run_Init(void){
    chThdCreateStatic(waADCLCD,          sizeof(waADCLCD),
        NORMALPRIO,          ADCLCD,          NULL);
    palSetPadMode(GPIOD,13,PAL_MODE_OUTPUT_PUS
        HPULL);
    palSetPadMode(GPIOD,15,PAL_MODE_OUTPUT_PUS
        HPULL);
    chThdCreateStatic(waBlink,          sizeof(waBlink),
        NORMALPRIO, Blink, NULL);
    chThdSleepMilliseconds(1000);
    chThdCreateStatic(waRECORD,          sizeof(waRECORD),
        NORMALPRIO, RECORD, NULL);
}

```

```

}
void Hasil_Adc(void){
    adc_co =adc0;
    adc_so2=adc1;
    adc_co2=adc2;
    adc_nox=adc3;

    v_co = (0.2279*adc_co)-32.492;
    v_so2 = (0.0018*adc_so2)-1.205;
    v_co2 = ((-0.2446*adc_co2)+926.94);
    v_nox = ((adc3* 9.9)/4095)*0.1;

    Lcd_Cursor(0,0);
    chprintf((BaseSequentialStream *)&myLCD,"co    =%7.1f
    ppm",v_co);
    Lcd_Cursor(0,1);
    chprintf((BaseSequentialStream *)&myLCD,"so2=%7.3f
    ppm",v_so2);
    Lcd_Cursor(0,2);
    chprintf((BaseSequentialStream *)&myLCD,"co2=%7.1f
    ppm",v_co2);
    Lcd_Cursor(0,3);
    chprintf((BaseSequentialStream *)&myLCD,"nox=%7.2f
    ppm",v_nox) ;
}

```

```

void Tulis_Adc(void){
    FIL FDLLogFile;
    memset(&FDLogFile, 0, sizeof(FIL));
    FRESULT err_file;
    UINT bw;
    char buffer[64];

    palSetPad(GPIOD, 15);
}

```

```

err_file = f_open(&FDLogFile, "Data Monitoring.csv",
    FA_WRITE | FA_OPEN_ALWAYS );
if (err_file == FR_OK || err_file == FR_EXIST){
    err_file = f_lseek(&FDLogFile, f_size(&FDLogFile));
    if(err_file == FR_OK){

        chsnprintf(buffer,64,"%2i-%2i-
        %4i;%2i:%2i;%4i;%4i;%4i;%4i\r\n",calendar.date,calenda
        r.month,calendar.year,calendar.hours,calendar.minutes,adc
        _co,adc_so2,adc_co2,adc_nox);
        f_write(&FDLogFile, buffer, strlen(buffer), &bw);
        f_close(&FDLogFile);
        chprintf((BaseSequentialStream *)&SDU1, "Some text
        written\r\n");
        return;
    }else{
        chprintf((BaseSequentialStream *)&SDU1, "Failed to
        seek file\r\n");
        return;
    }
}
}
}
}
}

```

//utama.h

```

#ifndef TA_UTAMA_H
#define TA_UTAMA_H
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include "ch.h"
#include "hal.h"

```



```
#include "chprintf.h"  
#include "memstreams.h"  
#include "chstreams.h"  
#include "ta_adc.h"  
#include "ta_lcd.h"  
#include "ta_rtc.h"  
#include "ta_mmc.h"  
#include "ta_uart.h"
```

```
void Run_Init(void);  
void Hasil_Adc(void);  
void Tulis_Adc(void);
```

```
#endif // TA_UTAMA_H
```


LAMPIRAN B (TABLE T STUDENT)

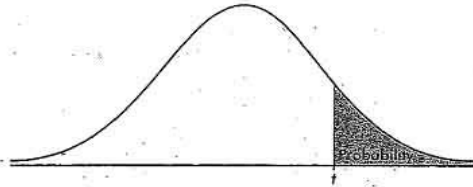
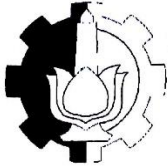


TABLE B: t-DISTRIBUTION CRITICAL VALUES

df	Tail probability <i>p</i>											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
Confidence level <i>C</i>												
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%

LAMPIRAN C

(SURAT KETERANGAN HASIL PENGAMBILAN DATA MONITORING GAS KARBONMONOKSIDA)



LABORATORIUM PENCEMARAN UDARA DAN PERUBAHAN IKLIM
DEPARTEMEN TEKNIK LINGKUNGAN
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

KAMPUS ITS SUKOLILO SURABAYA
TELPON (031)5948886, FAX. (031)5928397

DATA KALIBRASI ALAT DETEKTOR PENCEMAR UDARA

Pengirim : Lailatul Mufida / 2414031062

Jenis Pencemar : Karbon Monoksida (CO)

Pengukuran	A D C	Standard (ppm)
1	284	32,30
2	273	29,76
3	262	27,20
4	255	25,80
5	246	22,70
6	217	17,90
7	213	16,46
8	205	14,19
9	196	11,27
10	189	10,94
11	173	6,81

Surabaya, 20 Juli 2017

Kepala Laboratorium Pencemaran Udara dan Perubahan Iklim
Departemen Teknik Lingkungan FTSP-ITS



Dr. H. Agus Purwati, ST, MEPM
NIP. 196404112005011001

LAMPIRAN D
(DATASHEET SENSOR GAS MQ7)



Toxic Gas Sensor
(Model: MQ-7)

Manual

Version: 1.3

Valid from: 2014-05-01

Zhengzhou Winsen Electronics Technology Co., Ltd

Statement

This manual copyright belongs to Zhengzhou Winsen Electronics Technology Co., LTD. Without the written permission, any part of this manual shall not be copied, translated, stored in database or retrieval system, also can't spread through electronic, copying, record ways.

Thanks for purchasing our product. In order to let customers use it better and reduce the faults caused by misuse, please read the manual carefully and operate it correctly in accordance with the instructions. If users disobey the terms or remove, disassemble, change the components inside of the sensor, we shall not be responsible for the loss.

The specific such as color, appearance, sizes &etc, please in kind prevail.

We are devoting ourselves to products development and technical innovation, so we reserve the right to improve the products without notice. Please confirm it is the valid version before using this manual. At the same time, users' comments on optimized using way are welcome.

Please keep the manual properly, in order to get help if you have questions during the usage in the future.

Zhengzhou Winsen Electronics Technology CO., LTD

MQ-7 Semiconductor Sensor for Carbon Monoxide

Profile

Sensitive material of MQ-7 gas sensor is SnO_2 , which with lower conductivity in clean air. It make detection by method of cycle high and low temperature, and detect CO at low temperature (heated by 1.5V). The sensor's conductivity gets higher along with the CO gas concentration rising. At high temperature (heated by 3.0V), it cleans the other gases adsorbed at low temperature. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit.

Features

It has good sensitivity to carbon monoxide in wide range, and has advantages such as long lifespan, low cost and simple drive circuit & etc.

Main Applications

It is widely used in domestic CO gas leakage alarm, industrial CO gas alarm and portable CO gas detector.

Technical Parameters

Table 1

Model			MQ-7
Sensor Type			Semiconductor
Standard Encapsulation			Plastic cap
Target Gas			carbon monoxide
Detection range			10~500ppm CO
Standard Circuit Conditions	Loop Voltage	V_L	$\leq 10\text{V DC}$
	Heater Voltage	V_H	5.0V $\pm 0.1\text{V AC or DC}$ (High tem.) 1.5V $\pm 0.1\text{V AC or DC}$ (Low tem.)
	Heater Time	T_H	60 S ± 5 (High tem.) 90 S ± 5 (Low tem.)
	Load Resistance	R_L	Adjustable
Sensor character under standard test conditions	Heater Resistance	R_H	290 ± 50 (room tem.)
	Heater consumption	P_H	$\leq 300\text{mW}$
	Sensitivity	S	$R_A(n\text{ air})/R_A(n\text{ 150ppm CO}) \geq 5$
	Output Voltage	V_S	2.5V~4.3V (in 150ppm CO)
Standard test conditions	Concentration Slope	α	$\leq 0.8(R_{100ppm}/R_{10ppm}, \text{CO})$
	Tem. Humidity		20 $^{\circ}\text{C}$ $\pm 2^{\circ}\text{C}$, 55% $\pm 5\%$ RH
	Standard test circuit		$V_L=5.0\text{V} \pm 0.1\text{V}$, V_H (High tem.): 5.0V $\pm 0.1\text{V}$, V_H (Low tem.): 1.5V $\pm 0.1\text{V}$
	Preheat time		Over 48 hours

NOTE: Output voltage (V_S) is V_{AS} in test environment.

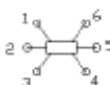
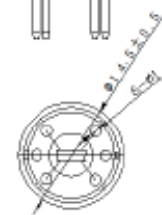
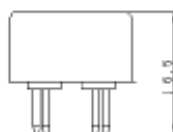


Fig1 Sensor Structure

Unit: mm

Basic Circuit

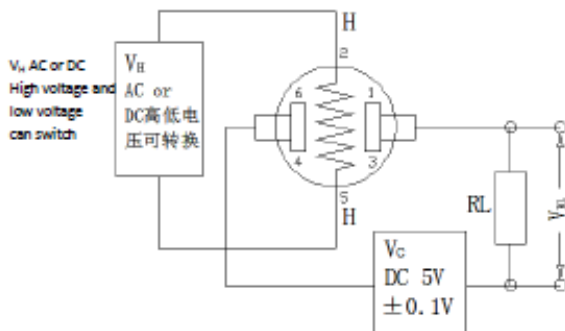


Fig2. MQ-7 Test Circuit

Instructions: The above fig is the basic test circuit of MQ-7. The sensor requires two voltage inputs: heater voltage (V_H) and circuit voltage (V_C). V_H is used to supply standard working temperature to the sensor and it can adopt DC or AC power. For this model sensor, V_H should be at $1.5V \pm 0.1V$ low voltage when detect CO while should be at $5V \pm 0.1V$ at non detection status (resuming period). V_H is the voltage of load resistance R_L which is in series with sensor. V_C supplies the detect voltage to load resistance R_L and it should adopts DC power.

Description of Sensor Characters

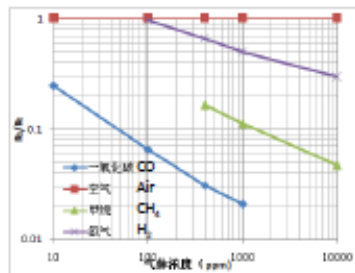


Fig3. Typical Sensitivity Curve

The ordinate is resistance ratio of the sensor (R_A/R_0), the abscissa is concentration of gases. R_A means resistance in target gas with different concentration, R_0 means resistance of sensor in clean air. All tests are finished under standard test conditions.

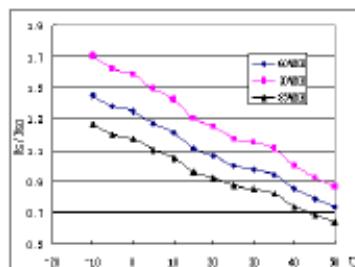


Fig4. Typical temperature/humidity characteristics

The ordinate is resistance ratio of the sensor (R_A/R_0). R_A means resistance of sensor in 150ppm CO gas under different tem. and humidity. R_0 means resistance of the sensor in 150ppm CO gas under 20°C/55%RH.

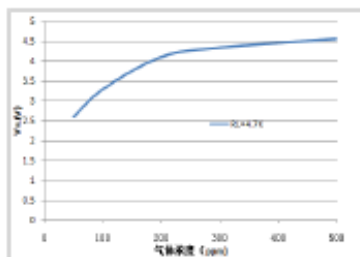


Fig5.Sensitivity Curve

Fig5 shows the V_{AS} in CO gas with different concentration. The resistance load R_L is 4.7 KΩ and the test is finished in standard test conditions.

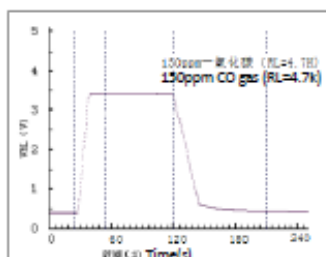


Fig6.Response and Resume

Fig5 shows the changing of V_{AS} in the process of putting the sensor into target gas and removing it out.

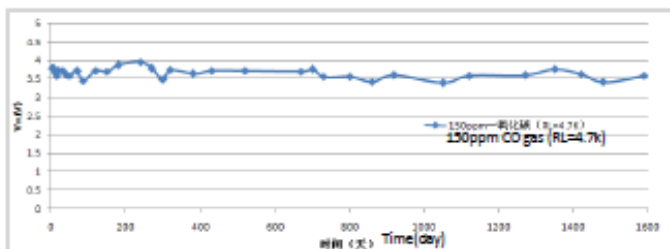


Fig7.Long-term Stability

Test is finished in standard test conditions, the abscissa is observing time and the ordinate is V_{AS} .

Cautions

1. Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Sensing material will lose sensitivity and never recover if the sensor absorbs organic silicon steam. Sensors must avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment.

1.2 High Corrosive gas

If the sensors are exposed to high concentration corrosive gas (such as H_2S , SO_2 , Cl_2 , HCl etc.), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt

1.4 Touch water

Sensitivity of the sensors will be reduced when splattered or dipped in water.

1.5 Freezing

Do avoid icing on sensor's surface, otherwise sensing material will be broken and lost sensitivity.

1.6 Applied higher voltage

Applied voltage on sensor should not be higher than stipulated value, even if the sensor is not physically damaged or broken, it causes down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, Pin 2&5 is heating electrodes, Pin (1,3)/(4,6) are testing electrodes (Pin 1 connects with Pin 3, while Pin 4 connects with Pin 6). If apply voltage on Pin 1&3 or 4&6, it will make lead broken; and no signal putout if apply on pins 2&4.



Fig8. Lead sketch

2. Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will influence sensors' performance lightly. However, if water condensation on sensors surface and keep a certain period, sensors' sensitive will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if it is placed in high gas concentration for long time, sensors' characteristic will be affected. If lighter gas sprays the sensor, it will cause extremely damage.

2.3 Long time storage

The sensors resistance will drift reversibly if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof bag without volatile silicon compound. For the sensors with long time storage but no electrify, they need long galvanical aging time for stability before using. The suggested aging time as follow:

Stable2.

Storage Time	Suggested aging time
Less than one month	No less than 48 hours
1 ~ 6 months	No less than 72 hours
More than six months	No less than 168 hours

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc., it will influence the sensors' performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then break. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage Conditions

2.7.1 For sensor, handmade welding is optimal way. The welding conditions as follow:

- Soldering flux: Rosin soldering flux contains least chlorine

- Homothermal soldering iron
 - Temperature: 250℃
 - Time: less than 3 seconds
- 2.7.1 If users choose wave-soldering, the following conditions should be obey:
- Soldering flux: Rosin soldering flux contains least chlorine
 - Speed: 1-2 Meter/Minute
 - Warm-up temperature: 100±20℃
 - Welding temperature: 250±10℃
 - One time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will reduce.

Zhengzhou Winsen Electronics Technology Co., Ltd
Add: No.299, Jinsuo Road, National Hi-Tech Zone,
Zhengzhou 450001 China
Tel: +86-371-67169097/67169670
Fax: +86-371-60932988
E-mail: sales@winsensor.com
Website: www.winsensor.com

BIODATA PENULIS



Penulis dilahirkan di Surabaya pada tanggal 21 April 1996 dengan diberi nama Lailatul Mufida. Bapak bernama M.Ghufron, Ibu bernama Siti Aminah, dan kakak laki-laki bernama Muhammad Tamim Aqsho dan Miftakhul Huda. Penulis telah menyelesaikan studi di SDN Siwalan Kerto II Surabaya pada tahun 2008, SMP Negeri 2 Krian Sidoarjo pada tahun 2011, SMA Negeri 1 Waru Sidoarjo pada tahun 2014, dan kemudian melanjutkan kuliah di Institut Teknologi Sepuluh Nopember (ITS), Departemen Teknik Instrumentasi, Program Studi D3 Teknik Instrumentasi, Fakultas Vokasi pada tahun 2014. Pengalaman organisasi yang pernah diikuti adalah menjadi staff BEM FTI-ITS, dan menjadi Asisten Laboratorium Pengukuran Fisis Teknik Fisika-ITS. Pengalaman magang (*on job training* / kerja praktek) di PT. Aneka Gas Industri (AGI), Sidoarjo, Jawa Timur dengan judul: Studi Sistem Pengendalian *Pressure* Pada Proses *Pressurize* Menggunakan *Pneumatic Control Valve* di *Molecular Sieve Tower* (MS Tower) PT. Aneka Gas Industri (AGI) SIER, Jawa Timur. Bagi pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini, dapat menghubungi penulis melalui nomor telepon 08979418906 atau email mufidahil34@gmail.com.